

Assignment 02

Yash Yadav(107)

Q1)

A screenshot of the Visual Studio Code interface. The left sidebar shows a project structure under 'WEB PROGRAMMING' with a folder 'Assignment 2' containing files like 'code.js', 'index.html', 'ass1.html', and '1st.js'. The main editor window displays the content of '1st.js'. The terminal at the bottom shows the output of running the script: a pattern of numbers from 1 to 9 repeated in a triangular shape. The status bar at the bottom right indicates the date as 11/20/2025.

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

```
PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node .\1st.js
1
2
3
4
5
6
7
8
9
```

Q2)

A screenshot of the Visual Studio Code interface, similar to the previous one but with a different script. The terminal window shows the output of running '1st.js', which prints the names 'Meena' and 'Veeva' because they have lengths greater than 4. The status bar at the bottom right indicates the date as 11/20/2025.

```
function nameLength(names) {
    for (let i = 0; i < names.length; i++) {
        if (names[i].length > 4) {
            console.log(names[i]);
        }
    }
}

let employees = ["Joy", "Meena", "Anne", "Xi", "Veeva"];
nameLength(employees);
```

```
PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node .\1st.js
Meena
Veeva
```

Q3)

The screenshot shows a code editor interface with a dark theme. At the top, there is a navigation bar with tabs for 'File', 'Go', 'Run', 'Terminal', and 'Help'. Below the navigation bar, there is a search bar with the text 'Web Programming'. The main area displays a file named '1st.js' with the following content:

```
Assignment 2 > JS 1stjs > ...
1  function min(arr) {
2
3      let minimum = arr[0];
4
5
6      for (let i = 1; i < arr.length; i++) {
7          if (arr[i] < minimum) {
8              minimum = arr[i];
9          }
10     }
11     return minimum;
12 }
13
14 let numbers = [100,99,65,44,9,88];
15 console.log(min(numbers));
16
17
```

Below the code editor, there is a terminal window with the following history:

- PS D:\CDAC Hyderabad\Web Programming\Assignment 2>
- PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node .\1st.js
- 9
- PS D:\CDAC Hyderabad\Web Programming\Assignment 2>

Q4)

```

index.html    ass1.html    JS 1stjs    JS greet.js    JS BLACKBOX
Assignment 2 > JS greet.js ...
1
2   function greet() {
3     const date = new Date();
4     const hours = date.getHours();
5
6     if (hours >= 5 && hours < 12) {
7       console.log("Good Morning");
8     }
9     else if (hours >= 12 && hours < 17) {
10      console.log("Good Afternoon");
11    }
12    else {
13      console.log("Good Evening");
14    }
15  }
16
17 module.exports = greet;
18

JS client.js ...
Assignment 2 > JS client.js ...
1
2   const greet = require("./greet");
3
4   greet();
5

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\VCDAC Hyderabad\Web Programming\Assignment 2> node .\client.js
Good Evening
PS D:\VCDAC Hyderabad\Web Programming\Assignment 2> []

OUTLINE TIMELINE

Trending videos Stranger Things...

Q5)

```

html    ass1.html    JS 1stjs    JS greet.js    JS calc.js    JS BLACKBOX
Assignment 2 > JS calc.js > add
1
2   function add(a, b) {
3     return a + b;
4   }
5
6   function subtract(a, b) {
7     return a - b;
8   }
9
10  function multiply(a, b) {
11    return a * b;
12  }
13
14  function divide(a, b) {
15    if (b === 0) return "Cannot divide by zero";
16    return a / b;
17  }
18
19  function square(a) {
20    return Math.pow(a, 2);
21  }
22
23  function min(a, b, c) {
24    return Math.min(a, b, c);
25  }
26
27  function max(a, b, c) {
28    return Math.max(a, b, c);
29  }
30
31  module.exports = { add, subtract, multiply, divide, square, min, max };

JS client.js ...
Assignment 2 > JS client.js ...
1
2   const calc = require('./calc');
3
4   console.log("Add: ", calc.add(2, 3));
5   console.log("Subtract: ", calc.subtract(10, 6));
6   console.log("Multiply: ", calc.multiply(10, 5));
7   console.log("Divide: ", calc.divide(10, 5));
8   console.log("Square: ", calc.square(5));
9   console.log("Minimum: ", calc.min(1, 5, 7));
10  console.log("Maximum: ", calc.max(10, 5, 7));


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\VCDAC Hyderabad\Web Programming\Assignment 2> node .\client.js
Add: 5
Subtract: 30
Multiply: 50
Divide: 2
Square: 25
Minimum: 1
Maximum: 34
PS D:\VCDAC Hyderabad\Web Programming\Assignment 2> []

OUTLINE TIMELINE

21C Mostly clear

Q6)

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it shows files like `calc.js`, `triangle.js`, `circle.js`, and `rectangle.js`.
- Code Editor:** There are four tabs open:
 - `JS calc.js`: Contains a function `isEquilateral(a, b, c)` which returns true if all sides are equal.
 - `JS triangle.js`: Contains functions `calArea`, `calCircumference`, `calDiameter`, and `calPerimeter`.
 - `JS circle.js`: Contains functions `calArea`, `calCircumference`, `calDiameter`, and `calPerimeter`.
 - `JS client.js`: A test script that imports the other modules and logs their results.
- Terminal:** At the bottom, the terminal shows the execution of `node .\client.js` and its output:

```
PS D:\XDCM hyderabad\Web Programming\Assignment> node .\client.js
Circle Area: 113.0973355292355
Circle Circumference: 50.26548245743669
Circle Diameter: 16
Rectangle Area: 24
Rectangle Perimeter: 26
Is Triangle Equilateral: true
Triangle Perimeter: 21
```
- Status Bar:** The status bar at the bottom right shows "Ln 17, Col 1" and "Port: 5500".

Q7)

The screenshot shows a Microsoft Edge browser window with a dark theme. The address bar at the top displays "Web Programming". The main content area is a file explorer interface. On the left, there's a sidebar with icons for file types like HTML, JS, CSS, and images. The main pane shows a folder structure under "Assignment 2". Inside "Assignment 2", there are files: "names.txt", "JS client.js", "JS greet.js", "JS triangle.js", "JS circles.js", and "index.html". "JS client.js" is currently selected and its code is visible:

```
const fs = require("fs");
var arr = ["Yash", "Jayprakash", "Yadav"];
var data = arr.join(" | ");
fs.writeFile("names.txt", data, (err) => {
  if (err) {
    console.log("Error writing file:", err);
  } else {
    console.log("File written successfully!");
  }
});
```

Below the code editor, there are tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". The "TERMINAL" tab is active, showing three command-line entries from a PowerShell session:

- PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node .\client.js
File written successfully!
- PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node .\client.js
File written successfully!
- PS D:\CDAC Hyderabad\Web Programming\Assignment 2> []

At the bottom of the screen, there's a taskbar with various pinned icons, including Microsoft Edge, File Explorer, and File Manager. The system tray shows the date and time as "8:41 PM 11/20/2023".

Q8)

A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project structure under 'WEB PROGRAMMING' with files like 'client.js', 'emp.txt', 'JS 1st.js', etc. The main editor window displays a Node.js script named 'client.js'. The script reads the contents of 'emp.txt' into 'data', splits it into lines, and then into parts separated by ':'. It calculates the total salary and logs the result. The terminal below shows the output: 'Total Salary = 364000'. A powershell window is also visible.

```
const fs = require("fs");
fs.readFile("emp.txt", "utf8", (err, data) => {
  if (err) {
    console.log("Error reading file:", err);
    return;
  }
  let lines = data.trim().split("\n");
  let total = 0;
  for (let line of lines) {
    let parts = line.split(":");
    let salary = parseInt(parts[parts.length - 1]);
    total += salary;
  }
  console.log("Total Salary = " + total);
});
```

```
Assignment 2 > emp.txt
1 1001:Yash Sales:99000
2 1002:Harsh Accounts:220000
3 1003:Ram TechSupport:45000
4
```

```
ps D:\VCDAC Hyderabad\Web Programming\Assignment 2> node .\client.js
Total Salary = 364000
ps D:\VCDAC Hyderabad\Web Programming\Assignment 2>
```

```
ZHC Mostly clear
```

```
powershell - Assignment 2 +> powershell Assignment 2
```

Q9)

A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project structure under 'WEB PROGRAMMING' with files like 'client.js', 'emp.txt', 'JS 1st.js', etc. The main editor window displays a Node.js script named 'client.js'. It defines an array of employee objects, loops through them to build a string of data separated by semicolons, and then writes this data to 'emp.txt'. The terminal below shows the output: 'Employee data saved successfully!'. A powershell window is also visible.

```
const fs = require("fs");
let employees = [
  { empId: 1001, empName: "Yash", dept: "Sales", salary: 99000 },
  { empId: 1002, empName: "Harsh", dept: "Accounts", salary: 220000 },
  { empId: 1003, empName: "Ram", dept: "TechSupport", salary: 45000 }
];
let fileData = "";
for (let emp of employees) {
  let line = `${emp.empId};${emp.empName};${emp.dept};${emp.salary}\n`;
  fileData += line;
}
fs.writeFile("emp.txt", fileData, (err) => {
  if (err) {
    console.log("Error writing file:", err);
  } else {
    console.log("Employee data saved successfully!");
  }
});
```

```
Assignment 2 > emp.txt
1 1001:Yash:Sales:99000
2 1002:Harsh:Accounts:220000
3 1003:Ram:TechSupport:45000
4
```

```
ps D:\VCDAC Hyderabad\Web Programming\Assignment 2> node .\client.js
Total Salary = 364000
ps D:\VCDAC Hyderabad\Web Programming\Assignment 2> node .\client.js
Employee data saved successfully!
ps D:\VCDAC Hyderabad\Web Programming\Assignment 2>
```

```
ZHC Mostly clear
```

```
powershell - Assignment 2 +> powershell Assignment 2
```

Q9)

VS Code screenshot showing a file structure under WEB PROGRAMMING / Assignment 2. The current file is client.js. The code reads 'emp.json' and logs customer details to the terminal.

```
Assignment 2 > JS client.js > fs.readFile("emp.json", "utf8") callback
1 const fs = require('fs');
2
3 // Read the JSON file
4 fs.readFile("emp.json", "utf8", (err, data) => {
5   if (err) {
6     console.log("Error reading file:", err);
7     return;
8   }
9
10  // Parse JSON
11  let customers = JSON.parse(data);
12
13  // Display customer details
14  customers.forEach(customer => {
15    console.log("Customer Name:", customer.custname);
16    console.log("Address:", customer.address);
17    console.log("Phone No:", customer.phone);
18    console.log("Rating:", customer.rating);
19    console.log("-----");
20  });
21});
```

The emp.json file contains:

```
[{"custname": "Anil Patil", "address": "Abd 123", "phone": "8877669988", "rating": 8}, {"custname": "Anita Kulkarni", "address": "A102, Highstreet", "phone": "99675456", "rating": 7}, {"custname": "Kavita Menon", "address": "B8203, Pune", "phone": "123456789", "rating": 9}]
```

Terminal output:

```
PS D:\VDCM Hyderabad\Web Programming\Assignment> node .\client.js
Customer Name: Anil Patil
Address: abd 123
Phone No: 8877669988
Rating: 8
-----
Customer Name: Anita Kulkarni
Address: A102, Highstreet
Phone No: 99675456
Rating: 7
-----
Customer Name: Kavita Menon
Address: B8203, Pune
Phone No: 123456789
Rating: 9
```

Q10)

VS Code screenshot showing a file structure under WEB PROGRAMMING / Assignment 2. The current file is client.js. The code reads 'emp.json', calculates discounts for books, and saves the results to book.txt.

```
Assignment 2 > JS client.js > fs.readFile("emp.json", "utf8") callback > books.forEach() callback
1 const fs = require('fs');
2
3 fs.readFile("emp.json", "utf8", (err, data) => {
4   if (err) {
5     console.log("Error reading file:", err);
6     return;
7   }
8
9   let books = JSON.parse(data);
10
11   let fileData = "";
12
13   books.forEach(book => {
14     let discount = book.price * 0.10;
15     let finalPrice = book.price - discount;
16
17     let line = `${book.bookid} ${book.name} | ${book.author} | ${book.price} | ${finalPrice}\n`;
18     fileData += line;
19   });
20
21   fs.writeFile("book.txt", fileData, (err) => {
22     if (err) {
23       console.log("Error writing file:", err);
24     } else {
25       console.log("Books saved successfully!");
26     }
27   });
28 });
29});
```

The emp.json file contains:

```
[{"bookid": 1001, "name": "Core Java", "author": "Cay Horstmann", "price": 450}, {"bookid": 1002, "name": "Python Basics", "author": "John Mark", "price": 270}, {"bookid": 1003, "name": "Node Essentials", "author": "Tom White", "price": 400}]
```

Terminal output:

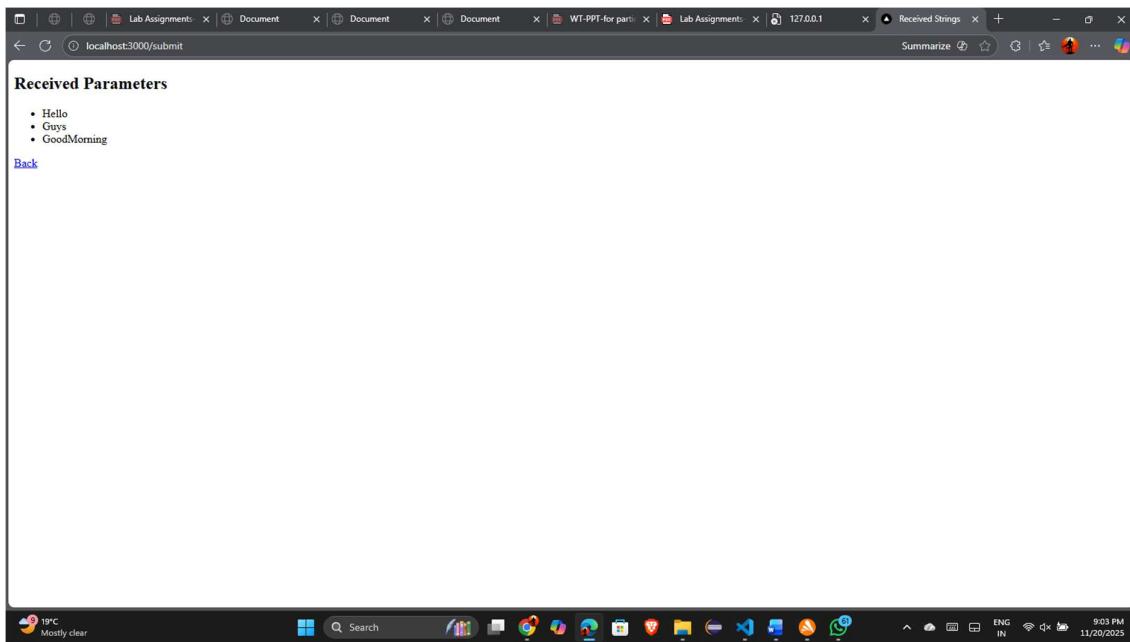
```
PS D:\VDCM Hyderabad\Web Programming\Assignment> node .\client.js
Books saved successfully!
PS D:\VDCM Hyderabad\Web Programming\Assignment>
```

Q11)

The screenshot shows the Visual Studio Code interface. On the left is the file explorer with a project structure for 'Assignment 2'. It includes files like 1st.js, book.txt, emp.txt, calc.js, circle.js, client.js, greet.js, names.txt, Rectangle.js, triangle.js, ass1.html, code.js, index.html, my-app, node_modules, and Practice. The 'client.js' file is open in the editor, displaying code that reads 'emp.txt' and logs each line to the console. The terminal at the bottom shows the command 'node .\client.js' being run, followed by the output: '1. 1. Node is a JS runtime platform', '2. 2. Created with C++ and Javascript', '3. 3. Runs on the V8 engine', and '4. 4. Created by Ryan Dahl'. The status bar at the bottom right indicates the port is 5500.

Q11)

The screenshot shows a web browser window with the URL 'localhost:3000'. The page contains a heading 'Enter three strings' and three input fields labeled 'String 1', 'String 2', and 'String 3'. Each field has the value 'Hello', 'Guys', and 'GoodMorning' respectively. Below the fields is a 'Send' button. The browser's address bar also shows '127.0.0.1' and a tab for 'Lab Assignment'. The status bar at the bottom right shows the date and time as 11/20/2025.



```
// app.js
const express = require('express');
const path = require('path');

const app = express();
const PORT = 3001;

// middleware to parse application/x-www-form-urlencoded
app.use(express.urlencoded({ extended: true }));

// serve index.html at root
app.get('/', (req, res) => {
    const str1, str2, str3 = res.body;
    // build XML response showing values in unordered list
    const html = `<head>
        <meta charset="utf-8" />
        <title>Received Strings</title>
    </head>
    <body>
        <h2>Received Parameters</h2>
        <ul>
            <li>${str1}</li>
            <li>${str2}</li>
            <li>${str3}</li>
        </ul>
        <a href="/">Back</a>
    </body>
</html>`;
    res.send(html);
});

// small helper to avoid HTML injection
function escapeHTML(str) {
    if (!str) return '';
    return String(str)
        .replace(/"/g, '"')
        .replace(/\&/g, '&lt;');
}

app.post('/submit', (req, res) => {
    const list = req.body;
    const str1 = escapeHTML(list['str1']);
    const str2 = escapeHTML(list['str2']);
    const str3 = escapeHTML(list['str3']);
    res.redirect(`?str1=${str1}&str2=${str2}&str3=${str3}`);
});

app.listen(PORT, () => console.log(`Listening on port ${PORT}`));

```

```
JS client.js          JS index.html Assignment 2 X  JS app.js          JS index.html Assignments   book.txt      emp.txt
Assignment 2 > 5 index.html ...
1  <!doctype html>
2  <html>
3  <head>
4  | <meta charset="utf-8" />
5  | <title>Send 3 Strings</title>
6  </head>
7  <body>
8  | <h2>Enter three strings:</h2>
9  | <form action="/submit" method="post">
10 |   <label>String 1: <input type="text" name="str1" required></label><br><br>
11 |   <label>String 2: <input type="text" name="str2" required></label><br><br>
12 |   <label>String 3: <input type="text" name="str3" required></label><br><br>
13 |   <button type="submit">Send</button>
14 | </form>
15 | </body>
16 | </html>
17 |
```

Q12)

The screenshot shows a web browser window with the URL `localhost:3000` in the address bar. The page title is "Simple Interest Calculator". It contains four input fields: "Principal Amount" with value "400000", "No. of Years" with value "5", "Rate of Interest (%)" with value "1", and a "Calculate" button. Below the form, the browser's developer tools are visible, showing the network tab with several requests listed.

Simple Interest Calculator

Principal Amount:

No. of Years:

Rate of Interest (%):

localhost:3000/calculate?p=400000&t=5&r=1

Simple Interest Result

Principal: 400000

Years: 5

Rate: 1%

Simple Interest = 20000

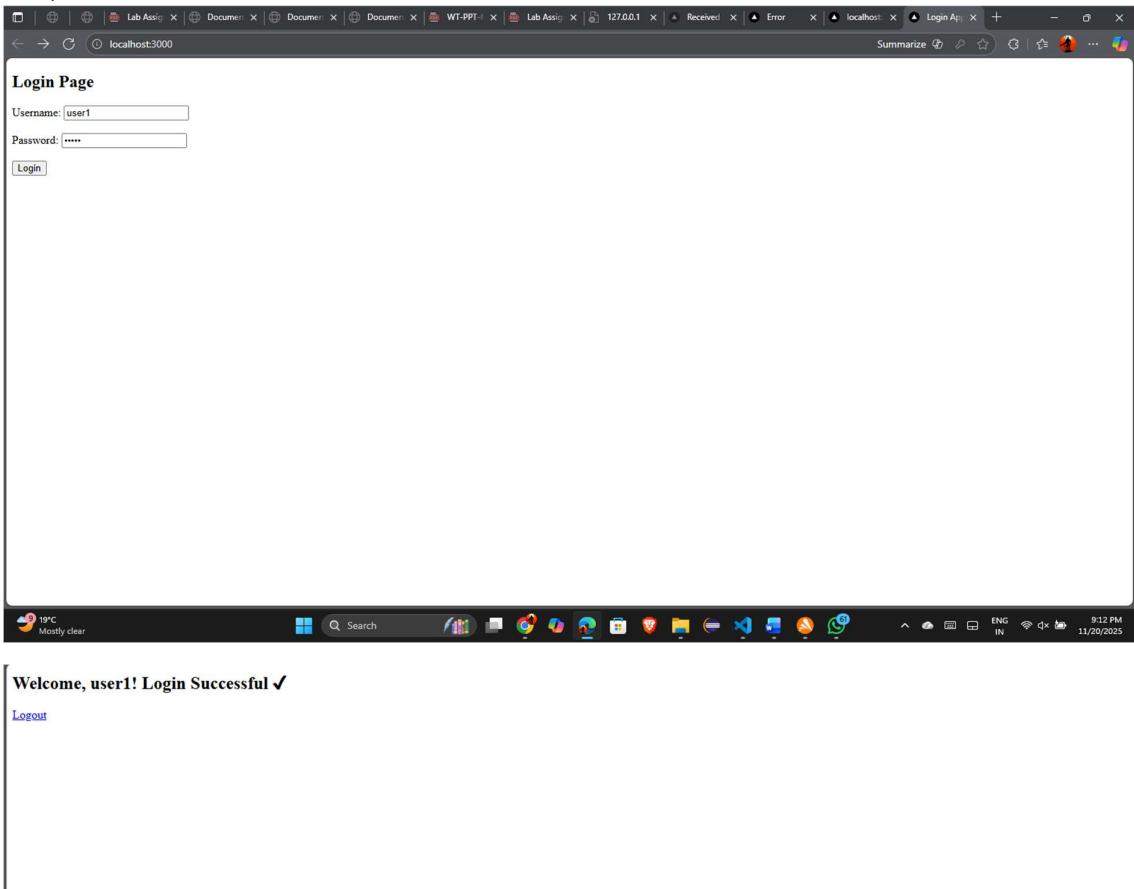
JS client.js index.html Assignment 2 JS app.js X index.html Assignments book.txt

```
Assignment 2 > JS app.js > app.get('/'). callback
1 const express = require('express');
2 const path = require('path');
3
4 const app = express();
5 const PORT = 3000;
6
7 // Serve HTML file
8 app.get('/', (req, res) => {
9   res.sendFile(path.join(__dirname, 'index.html'));
10 });
11
12 // Calculate Simple Interest
13 app.get('/calculate', (req, res) => {
14   const p = parseFloat(req.query.p);
15   const t = parseFloat(req.query.t);
16   const r = parseFloat(req.query.r);
17
18   const si = (p * t * r) / 100;
19
20   res.send(`
21     <h2>Simple Interest Result</h2>
22     <p>Principal: ${p}</p>
23     <p>Years: ${t}</p>
24     <p>Rate: ${r}%</p>
25     <h3>Simple Interest = ${si}</h3>
26     <a href="/">Go Back</a>
27   `);
28 });
29
30 app.listen(PORT, () => {
31   console.log(`Server running at http://localhost:${PORT}`);
32 });
33
```

JS client.js index.html Assignment 2 X JS app.js index.html Assignments

```
Assignment 2 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Simple Interest Calculator</title>
6  </head>
7  <body>
8    <h2>Simple Interest Calculator</h2>
9
10   <form action="/calculate" method="get">
11     <label>Principal Amount: </label>
12     <input type="number" name="p" required><br><br>
13
14     <label>No. of Years: </label>
15     <input type="number" name="t" required><br><br>
16
17     <label>Rate of Interest (%): </label>
18     <input type="number" name="r" required><br><br>
19
20     <button type="submit">Calculate</button>
21   </form>
22 </body>
23 </html>
```

Q13)



```
JS client.js      index.html Assignment 2      JS app.js      index.html Assignments      book.txt      emp.txt
Assignment 2 > JS app.js > ↗ app.get('/') callback
1  const express = require('express');
2  const path = require('path');
3  const app = express();
4
5  const PORT = 3000;
6
7  // User database (array)
8  let users = [
9    { uname: "shrilata", pass: "secret" },
10   { uname: "admin", pass: "admin123" },
11   { uname: "user1", pass: "pass1" }
12 ];
13
14 // Serve Login page
15 app.get('/', (req, res) => {
16   res.sendFile(path.join(__dirname, 'index.html'));
17 });
18
19 // Handle Login request
20 app.get('/login', (req, res) => {
21   const uname = req.query.uname;
22   const pass = req.query.pass;
23
24   // Check if user exists
25   const found = users.find(u => u.uname === uname && u.pass === pass);
26
27   if (found) {
28     res.send(`<h2>Welcome, ${uname}! Login Successful ✓</h2>
29             <a href="/">Logout</a>`);
30   } else {
31     res.send(`<h2 style="color:red;">Invalid Username or Password ✗</h2>
32             <a href="/">Try Again</a>`);
33   }
34 });
35
36 // Start server
37 app.listen(PORT, () => {
38   console.log(`Server running at http://localhost:${PORT}`);
39 });
40
```

```
JS client.js      index.html Assignment 2      JS app.js      index.html Assignments      book.txt
Assignment 2 > ↗ index.html > ↗ html > ↗ body > ↗ form
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Login App</title>
6  </head>
7  <body>
8    <h2>Login Page</h2>
9
10   <form action="/login" method="get">
11     <label>Username: </label>
12     <input type="text" name="uname" required><br><br>
13
14     <label>Password: </label>
15     <input type="password" name="pass" required><br><br>
16
17     <button type="submit">Login</button>
18   </form>
19 </body>
20 </html>
21
```

Q14)

Signup Form

Username: Yash

Password: *****

Signup

localhost:3000/signup

Signup Successful ✓

User Added : Yash

[Go Back](#)

```
Assignment 2 > JS app.js > index.html Assignment 2
```

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

// For handling POST data
app.use(express.urlencoded({ extended: true }));

let users = [];

// Serve static HTML page
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'index.html'));
});

// Handle signup
app.post('/signup', (req, res) => {
  const uname = req.body.uname;
  const pass = req.body.pass;

  let userobj = {
    uname: uname,
    pass: pass
  };

  // Add to users array
  users.push(userobj);

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

  res.send(`
    <h2>Signup Successful ✓</h2>
    <p>Redirecting to home</p>
    <a href="/">Go Back</a>
  `);
});

// start server
app.listen(8080, () => {
  console.log("Server running at http://localhost:8080");
});
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\VDCM HyderabadWeb Programming\Assignment 2> node app.js

Users Array: [{ uname: 'Yash', pass: 'Yadav' }]

Users Array: [{ uname: 'Yash', pass: 'Yadav' }, { uname: 'Yash', pass: 'Yadav' }]

LN 12, COL 45 SPACES: 4 UTF-8 CRLF ↵ JavaScript ⚡ Port: 5500 ⚡ Prettier ⚡

19°C Mostly clear

File Edit Selection View Go Run Terminal Help

Assignment 2 > index.html Assignment 2 JS app.js index.html Assignments book.txt emp.txt

WEB PROGRAMMING

- JS client.js
- JS 1st.js
- JS app.js
- book.txt
- JS calc.js
- JS circle.js
- JS client.js
- JS emp.txt
- JS greet.js
- index.html
- names.txt
- JS Rectangle.js
- JS triangle.js

Assignments

- Assig1.html
- JS code.js
- index.html

my-app

node_modu...

Practice

package-loc..

package.json

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Signup</title>
</head>
<body>
    <form action="/sign" method="post">
        <label>Username:</label>
        <input type="text" name="uname" required><br><br>
        <label>Password:</label>
        <input type="password" name="pass" required><br><br>
        <button type="submit">Signup</button>
    </form>
</body>
</html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node app.js

Users Array: [{ uname: 'Yash', pass: 'Yadav' }]

Users Array: [{ uname: 'Yash', pass: 'Yadav' }, { uname: 'Yash', pass: 'Yadav' }]

Ln 21, Col 1 Spaces: 4 UTF-8 CRLF HTML ENG Port: 5500 ✎ Prettier

19°C Mostly clear

Search

OUTLINE TIMELINE

Node Assignment 2 powershell

Port: 5500 9:17 PM

Q15)

File Edit Selection View Go Run Terminal Help

Assignment 2 > index.html Assignment 2 JS app.js index.html Assignments book.txt emp.txt

WEB PROGRAMMING

- JS client.js
- JS 1st.js
- JS app.js
- book.txt
- JS calc.js
- JS circle.js
- JS client.js
- JS emp.txt
- JS greet.js
- index.html
- names.txt
- JS Rectangle.js
- JS triangle.js

Assignments

- Assig1.html
- JS code.js
- index.html

my-app

node_modu...

Practice

package-loc..

package.json

```
const express = require('express');
const app = express();
const PORT = 3000;

let studs = [
    { sid: 101, sname: "swathi", course: "mca" },
    { sid: 102, sname: "swetha", course: "mca" },
    { sid: 103, sname: "Anita", course: "DESO" },
    { sid: 104, sname: "sunitha", course: "DEOI" },
    { sid: 105, sname: "Rabitha", course: "DWEC" }
];

// RESTful API /students/:name
app.get('/students/:name', (req, res) => {
    let name = req.params.name;
    // Search student by name (case-insensitive)
    let found = studs.filter(stu =>
        stu.sname.toLowerCase() === name.toLowerCase()
    );
    if (found.length > 0) {
        res.send(found);
    } else {
        res.send({ message: "Student not found!" });
    }
});

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

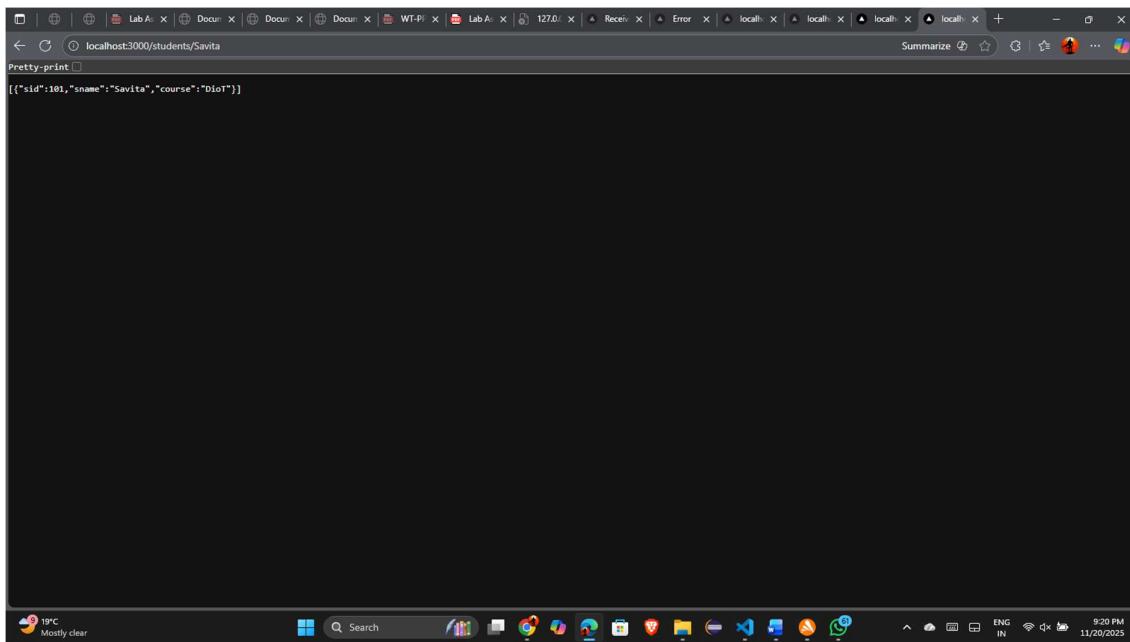
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node app.js

>>> Server running at http://localhost:3000

Ln 34, Col 1 Spaces: 4 UTF-8 CRLF JavaScript ENG IN 9:18 PM 11/20/2025

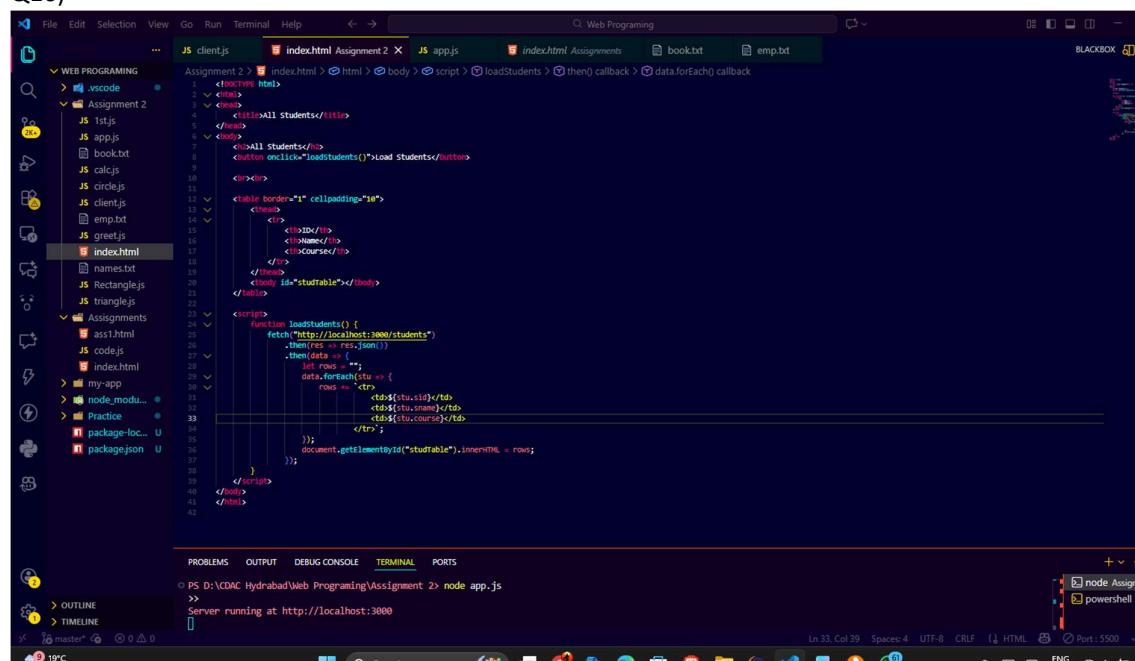
Node Assignment 2 powershell



A screenshot of a Microsoft Edge browser window. The address bar shows 'localhost:3000/students/Savita'. The developer tools console tab is open, with the 'Pretty-print' checkbox checked. The console displays the following JSON array:

```
[{"sid": "101", "sname": "Savita", "course": "D101"}]
```

Q16)



A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project structure under 'WEB PROGRAMMING' with files like client.js, index.html, app.js, etc. The main editor area shows the 'index.html' file with the following content:

```
<!DOCTYPE html>
<html>
<head>
<title>All Students</title>
</head>
<body>
<h2>All Students</h2>
<button onclick="loadStudents()">Load Students</button>
<br><br>
<table border="1" cellpadding="10">
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Course</th>
</tr>
</thead>
<tbody id="studTable"></tbody>
</table>
<script>
function loadStudents() {
fetch("http://localhost:3000/students")
.then(data => data.json())
.then(data => {
let rows = "";
data.forEach(stu => {
rows += "<tr>";
rows += "<td>" + stu.sid + "</td>";
rows += "<td>" + stu.sname + "</td>";
rows += "<td>" + stu.course + "</td>";
rows += "</tr>";
});
document.getElementById("studTable").innerHTML = rows;
})
}
</script>
</body>
</html>
```

The bottom status bar shows the terminal output: 'PS D:\CDAC Hyderabad\Web Programming\Assignment 2> node app.js' and 'Server running at http://localhost:3000'. The bottom right corner shows the system tray with icons for battery, signal, and date/time.

File Edit Selection View Go Run Terminal Help

JS client.js JS app.js index.html Assignment 2

WEB PROGRAMMING

- Assignment 2
 - JS client.js
 - JS app.js
 - index.html
 - Assignment 2
 - JS 1st.js
 - JS calc.js
 - JS circle.js
 - JS client.js
 - JS emp.txt
 - JS greet.js
 - JS index.html
 - JS names.txt
 - JS Rectangle.js
 - JS triangle.js
 - Assignments
 - Assignment 2
 - JS code.js
 - JS index.html
 - my app
 - node module
 - Practice
 - package-lock.json
 - package.json

```

Assignment 2 > JS app.js > ...
1 const express = require('express');
2 const app = express();
3 const path = require('path');
4 const port = 3000;
5
6 let studs = [
7   { sid: 101, sname: "Savita", course: "DioT" },
8   { sid: 102, sname: "Kavita", course: "DAC" },
9   { sid: 103, sname: "Anita", course: "DESD" },
10  { sid: 104, sname: "Sunita", course: "DioT" },
11  { sid: 105, sname: "Babita", course: "DMC" }
12];
13
14 // Serve static HTML files
15 app.use(express.static(__dirname));
16
17 // REST Endpoint 1 = ALL students
18 app.get('/students', (req, res) => {
19   res.json(studs);
20 });
21
22 // REST Endpoint 2 = Search by name
23 app.get('/studentByName', (req, res) => {
24   let name = req.params.sname.toLowerCase();
25   let found = studs.filter(s => s.sname.toLowerCase() === name);
26
27   if (found.length > 0) {
28     res.json(found);
29   } else {
30     res.json({ message: "Student not found" });
31   }
32 });
33
34 app.listen(PORT, ()=> {
35   console.log(`Server running at http://localhost:${PORT}`);
36 });
37

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\VDAC\HyderabadWeb Programming\Assignment 2> node app.js

Server running at http://localhost:3000

19°C Mostly clear

localhost:3000

All Students

ID	Name	Course
101	Savita	DioT
102	Kavita	DAC
103	Anita	DESD
104	Sunita	DioT
105	Babita	DMC

node Assignment 2

powershell

LN 37, Col 1 Spaces: 4 UFT-8 CRLF JavaScript Port 5500 Prettier

9:24 PM 11/20/2025

19°C Mostly clear

Search

127.0.0.1 | Recd | Error | local | local | local | local | All S | Summarize

ENG IN 9:23 PM 11/20/2025