🧑‍💻 Day 11 Lab Sheet – Node.js Basics, npm & Server Creation

Objective

By the end of this lab, you will:

- Run JavaScript code with Node.js
- Use **core**, **local**, and **third-party** modules
- Manage packages using **npm**
- Build a **basic web server** using Node.js

Setup

1. Make sure Node.js is installed:
2. node -v
3. npm -v

✅ If versions appear, you’re ready to go.

4. Create a folder named Day11_NodeJS.
   Navigate into it using:
5. cd Day11_NodeJS

Task 1: Hello Node.js

1. Create a file hello.js:
2. console.log("Hello from Node.js!");
3. Run it:
4. node hello.js

✅ Output should print in the terminal.

Task 2: Exploring Core Modules

(a) Using the os module

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


console.log("Operating System:", os.platform());
console.log("Architecture:", os.arch());
console.log("Free Memory:", os.freemem());
console.log("Total Memory:", os.totalmem());
```

✅ Displays system information.

(b) Using the path module

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


console.log("File Name:", path.basename(__filename));
console.log("Directory:", path.dirname(__filename));
console.log("Extension:", path.extname(__filename));
```

✅ Shows file details using Node’s built-in module.

Task 3: Creating & Importing a Local Module

1. Create a file math.js:
2. function add(a, b) {
3.   return a + b;
4. }
5. function subtract(a, b) {
6.   return a - b;
7. }
8. function multiply(a, b) {
9.   return a * b;
10. }
11.
12. module.exports = { add, subtract, multiply };
13. Create app.js:
14. const math = require('./math');
15.
16. console.log("Addition:", math.add(5, 3));
17. console.log("Subtraction:", math.subtract(10, 4));
18. console.log("Multiplication:", math.multiply(6, 2));
19. Run:

20. node app.js

✅ Displays the results of all operations.

Task 4: Using npm & Third-Party Modules

1. Initialize npm:
2. npm init -y
3. Install a package (example: chalk):
4. npm install chalk
5. Create color.js:
6. const chalk = require('chalk');
7.
8. console.log(chalk.blue('This is a blue message'));
9. console.log(chalk.green('This is a green message'));
10. console.log(chalk.red.bold('This is a bold red message'));
11. Run:
12. node color.js

✅ Terminal shows colored text output.

Task 5: Creating a Basic HTTP Server

1. Create server.js:
2. const http = require('http');
3.
4. const server = http.createServer((req, res) => {
5. res.writeHead(200, { 'Content-Type': 'text/plain' });
6. res.end('Hello from Node.js Server!');
7. });
8.
9. server.listen(3000, () => {
10. console.log('Server running at http://localhost:3000/');
11. });
12. Run the server:

13. node server.js

14. Open browser → go to http://localhost:3000.
    ✅ “Hello from Node.js Server!” appears in the browser.

Task 6: Serving Simple HTML Content

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

http.createServer((req, res) => {
 res.writeHead(200, { 'Content-Type': 'text/html' });
 res.write('<h1>Welcome to My Node.js Server</h1>');
 res.end();
}).listen(4000);

console.log('Server running at http://localhost:4000/');
```

✅ Visit http://localhost:4000 → see HTML output.

Task 7: Basic Routing

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

http.createServer((req, res) => {
 res.writeHead(200, { 'Content-Type': 'text/plain' });

 if (req.url === '/') {
  res.end('Home Page');
 } else if (req.url === '/about') {
  res.end('About Page');
 } else if (req.url === '/contact') {
  res.end('Contact Page');
 } else {
  res.end('404 Not Found');
 }
}).listen(5000);

console.log('Server running at http://localhost:5000/');
```

✅ Displays different responses for /, /about, /contact.

Task 8: Bonus – Display Date & Time

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

http.createServer((req, res) => {
  let date = new Date();
  res.writeHead(200, { 'Content-Type': 'text/html' });
  res.end(`<h2>Current Time: ${date.toLocaleTimeString()}</h2>`);
}).listen(6000);

console.log('Server running at http://localhost:6000/');
```

✅ Shows real-time clock in browser.

✅ Deliverables

- hello.js
- math.js & app.js (local module)
- color.js (npm package)
- server.js, routing.js (basic servers)

All files should run correctly using Node and produce expected outputs.

💡 Optional Challenge

👉 Create a Node.js server that serves **different HTML files** for /, /about, and /contact.
(Hint: use the fs module to read HTML files dynamically.)