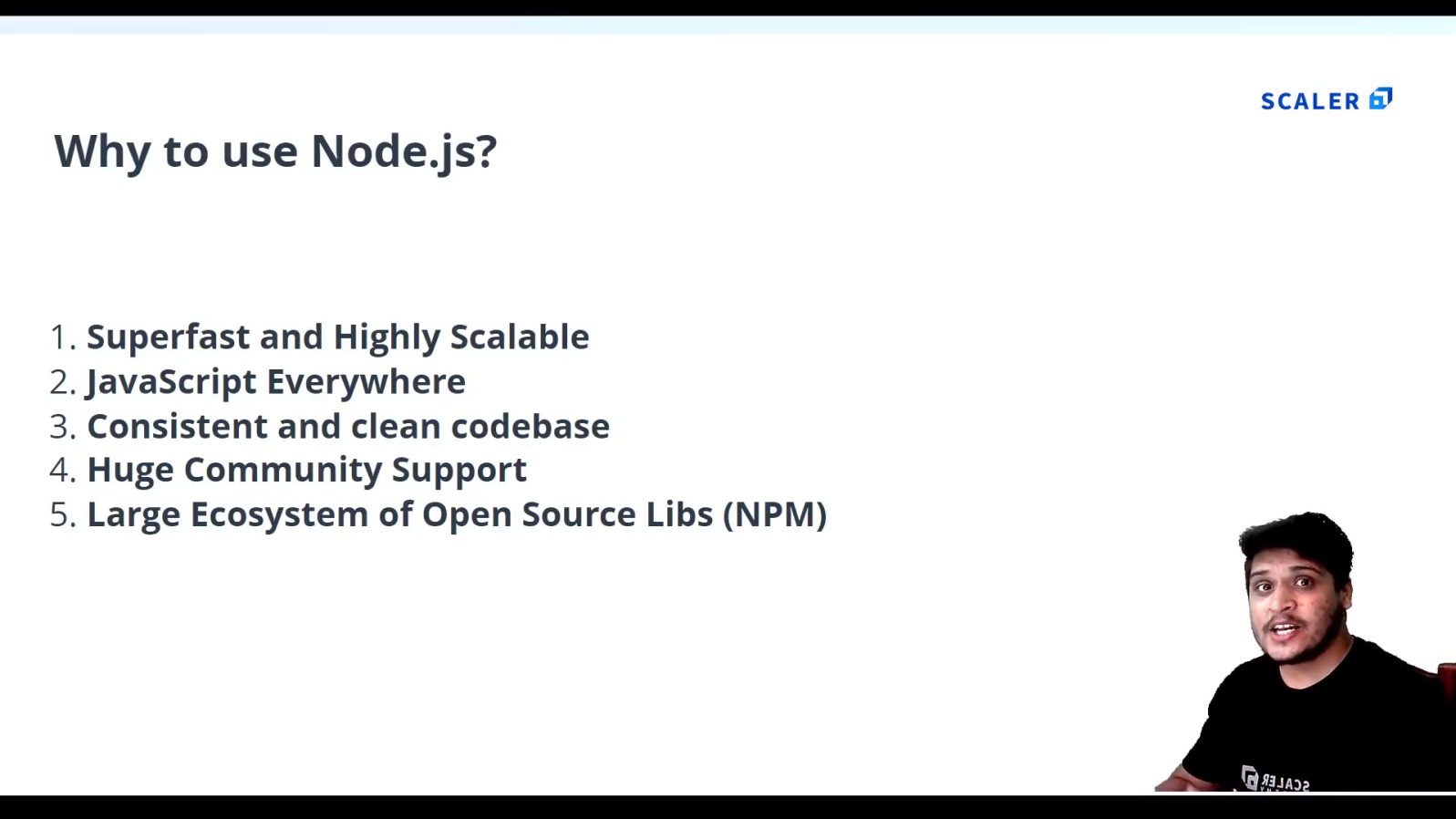
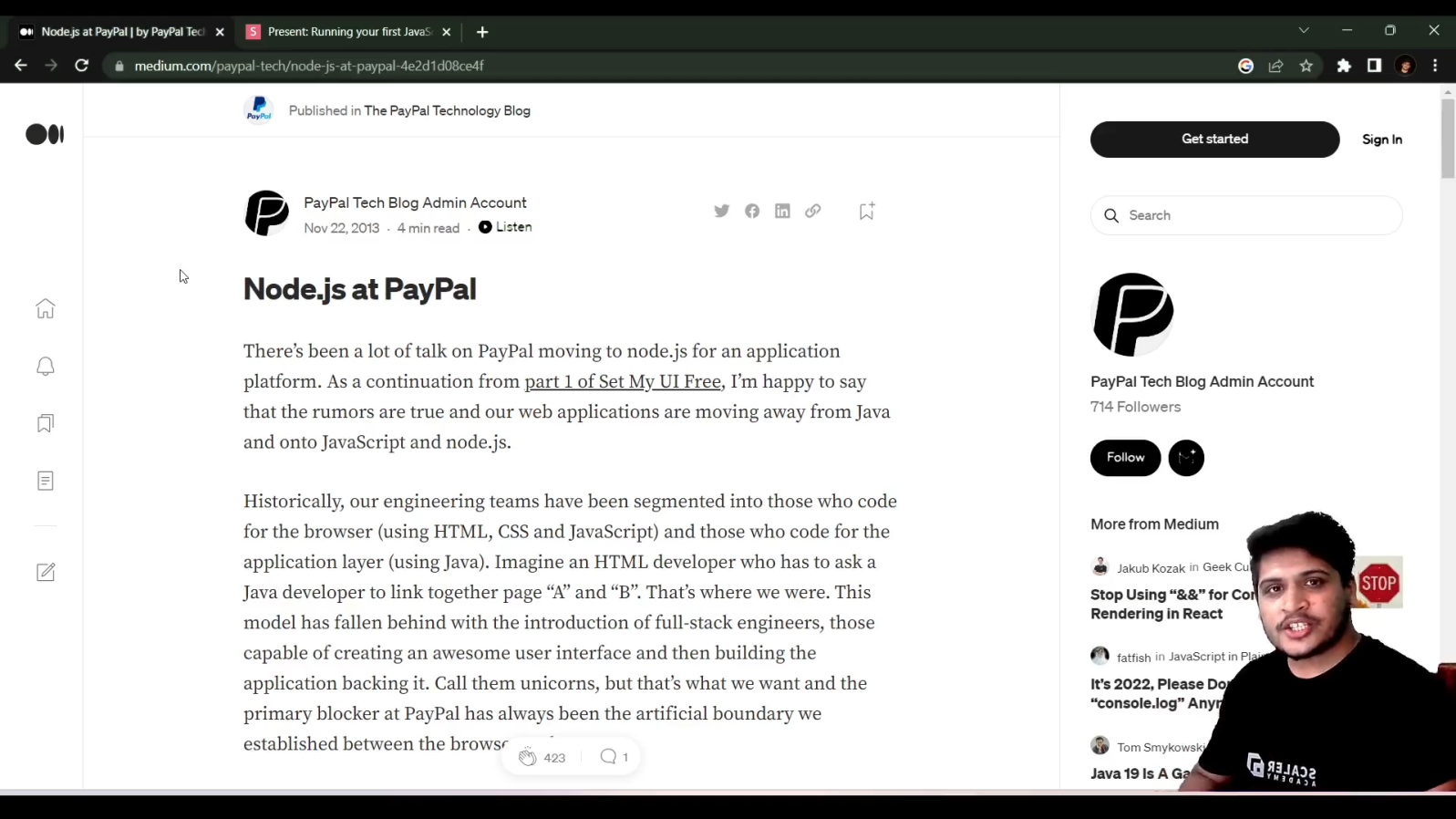
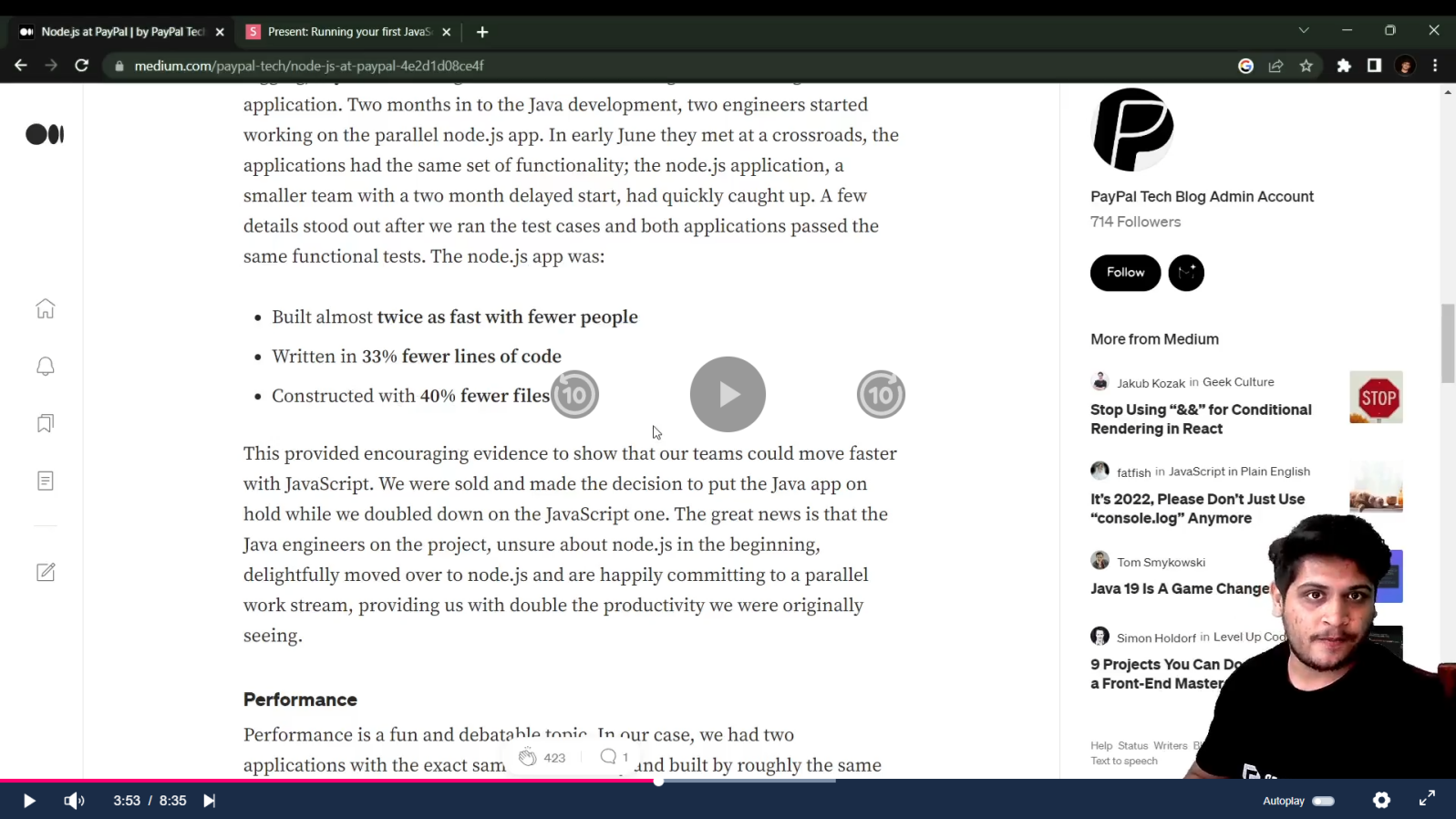
**NODE JS - SCALAR**

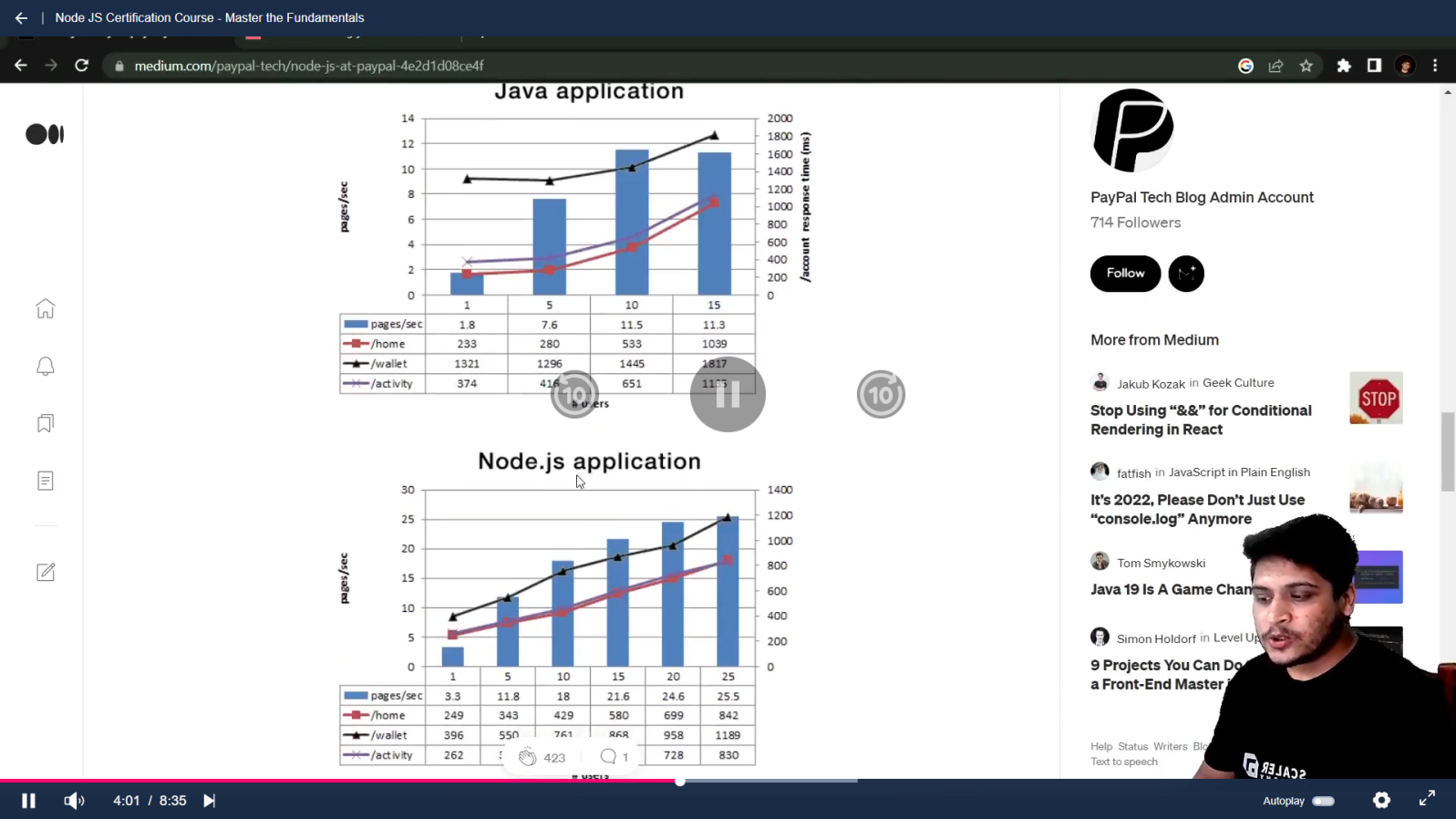
1. **Introduction :**

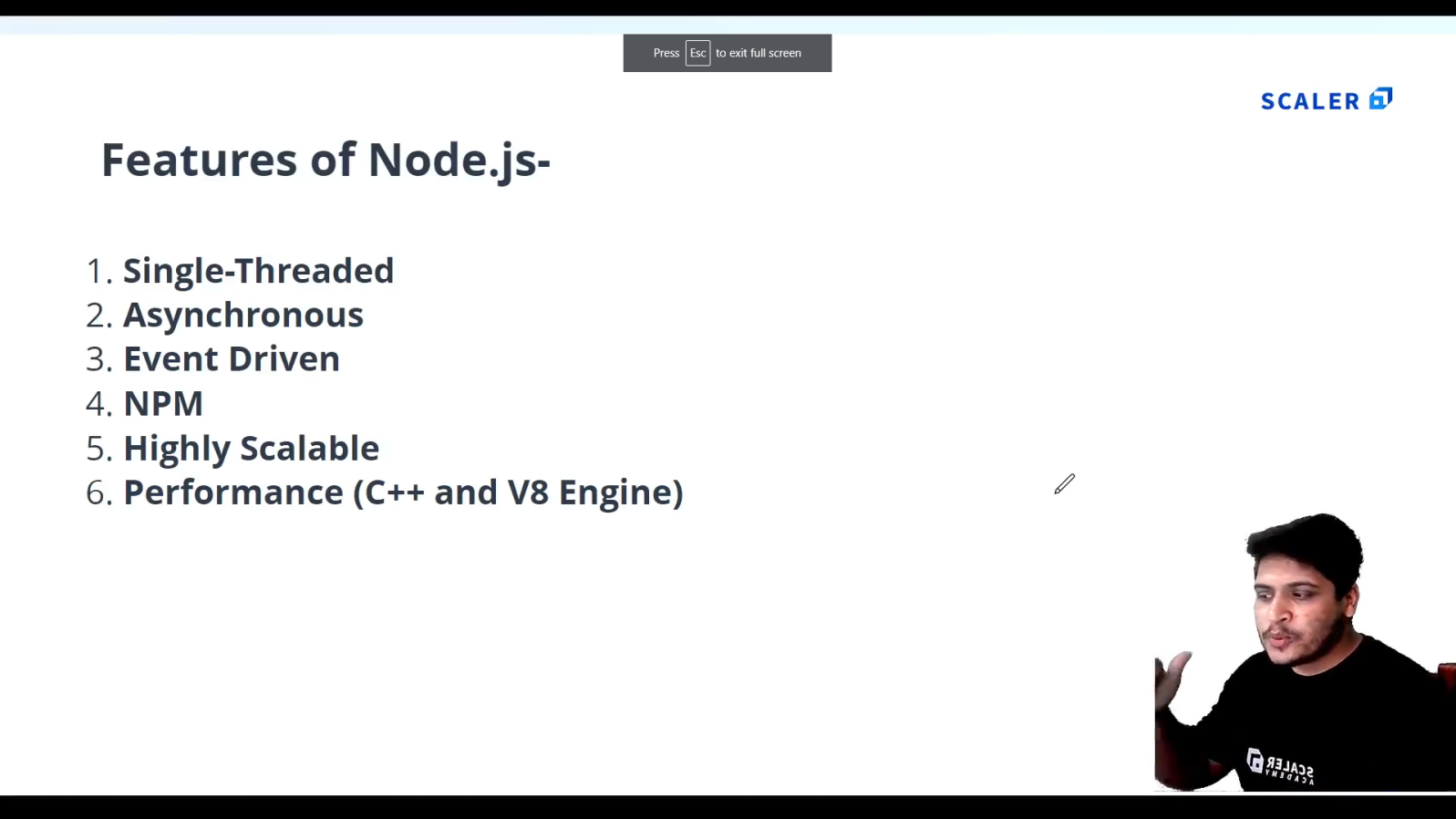
****

****

****

****

****

****

1. **Install Node.js and VScode.**
2. **First Node program :**
   1. **Create a js file with a function:p1\_test.js**

console.log('Hello World');

function sayHello(){

    console.log('Hello from anshad');

}

sayHello();

* 1. **To Execute type “node name\_of\_program.js” : node p1\_test.js**

1. **There is no “window” object in Node js instead we have “global” object.**

console.log(global);

PS E:\MCA\COURSES\NODE JS\nodejs\_certification@scalar> node p1\_test.js

<ref \*1> Object [global] {

  global: [Circular \*1],

  clearImmediate: [Function: clearImmediate],

  setImmediate: [Function: setImmediate] {

    [Symbol(nodejs.util.promisify.custom)]: [Getter]

  },

  clearInterval: [Function: clearInterval],

  clearTimeout: [Function: clearTimeout],

  setInterval: [Function: setInterval],

  setTimeout: [Function: setTimeout] {

    [Symbol(nodejs.util.promisify.custom)]: [Getter]

  },

  queueMicrotask: [Function: queueMicrotask],

  structuredClone: [Function: structuredClone],

  atob: [Getter/Setter],

  btoa: [Getter/Setter],

  performance: [Getter/Setter],

  fetch: [Function: value],

  crypto: [Getter]

}

1. **Node Module system:**
   1. **The “global” Object :**

//GLOBAL OBJECT

//are built-in objects that are part of the JavaScript and can be used directly in the application without importing any particular module.

let name = 'Anshad';

console.log(global.name);//undefined

* 1. **Modules and Modularity :**

1. **Create ‘calculator.js’ file :**

//CALCULATOR :

function add(a , b ){

    console.log( a + b );

}

function sub(a , b ){

    console.log( a - b );

}

function mul(a , b ){

    console.log( a \* b );

}

function div(a , b ){

    console.log( a / b );

}

//Exporting functions:

module.exports = {

    addition : add ,

    subtraction : sub,

    multiplication : mul,

    division : div

}

1. **Create main file ‘modularity.js’ file :**

//1.create a seperate file 'calculator.js'.

//Modularity lets us use those contents in calculator.js in this file.

const calculator = require('./calculator');//import calculator.js

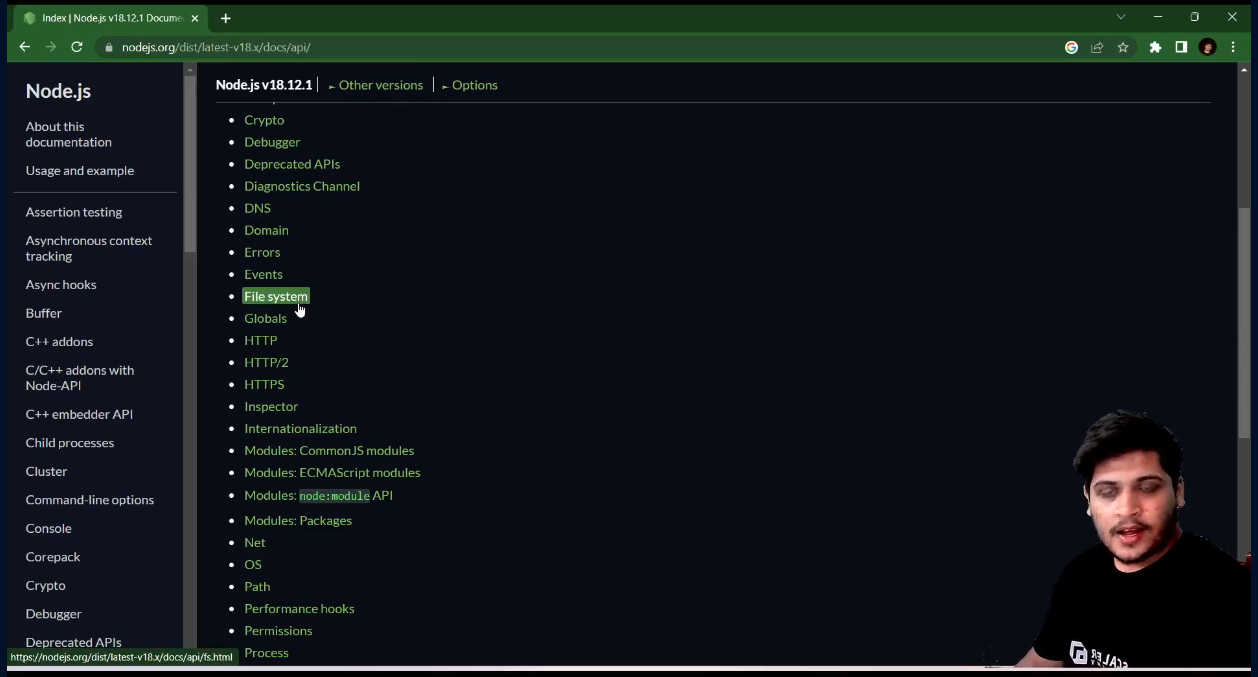
calculator.addition(3 , 4);//Calls the function in calculator.js by passing values to that.

calculator.subtraction( 5 ,2);

calculator.multiplication( 3 , 4);

calculator.division(10 , 2);

* 1. **Introduction to Node Modules :**



* 1. **Child Process Module :**

//Child process is a node module used to create sub process within a script .

//You can perform different tasks with your script by just using some methods.

const cp = require('child\_process');//Importing child-process module.

cp.execSync('calc') ; //Opens a calculator.['calc' is short code for opening calculator.]

cp.execSync('start chrome'); //opens chrome browser.

cp.execSync('start chrome https://www.google.com/');//opens google page directly

console.log('Output '+cp.execSync('node p1\_test.js'));//prints output from a specified File.

* 1. **OS Module :**

|  |  |
| --- | --- |
| **Code** | **Output** |
| //OS Module :(to get the information of your current system)  const os = require('os');//Import OS module  console.log(os.arch());//Displays Architecture of your OS (64 or 32)  console.log(os.platform());//Displays the platform of your OS  console.log(os.networkInterfaces());//Displays the network informations of your Os  console.log(os.cpus());//Displays the cpu's details (graphics,processor,configurations)  console.log('Total Memory : ',os.totalmem());//Displays the total memory .  console.log('Free Memory : ',os.freemem());//Displays the free memory. | PS E:\MCA\COURSES\NODE\_JS\nodejs\_certification@scalar\Module\_3> node .\osModule.js  x64  win32  {  Ethernet: [  {  address: '192.168.1.12',  netmask: '255.255.255.0',  family: 'IPv4',  mac: 'f8:0d:ac:29:a8:91',  internal: false,  cidr: '192.168.1.12/24'  }  ],  'Loopback Pseudo-Interface 1': [  {  address: '::1',  netmask: 'ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff',  family: 'IPv6',  mac: '00:00:00:00:00:00',  internal: true,  cidr: '::1/128',  scopeid: 0  },  {  address: '127.0.0.1',  netmask: '255.0.0.0',  family: 'IPv4',  mac: '00:00:00:00:00:00',  internal: true,  cidr: '127.0.0.1/8'  }  ]  }  [  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 101765, nice: 0, sys: 115328, idle: 1025421, irq: 13640 }  },  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 73468, nice: 0, sys: 51812, idle: 1117078, irq: 1656 }  },  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 94734, nice: 0, sys: 97968, idle: 1049656, irq: 1468 }  },  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 81906, nice: 0, sys: 47234, idle: 1113218, irq: 1312 }  },  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 58578, nice: 0, sys: 38953, idle: 1144828, irq: 1046 }  },  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 59406, nice: 0, sys: 34625, idle: 1148328, irq: 859 }  },  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 55937, nice: 0, sys: 33125, idle: 1153296, irq: 843 }  },  {  model: 'AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx ',  speed: 2096,  times: { user: 61718, nice: 0, sys: 31375, idle: 1149265, irq: 875 }  }  ]  Total Memory : 14901878784  Free Memory : 6341722112 |

* 1. **Path Module :**
  2. **Asds**
  3. **Asd**
  4. **asd**

1. **A**
2. **A**
3. **a**