**Html Self Prepared Notes**

**(08/04/2024)**

**HTML:**

- it is use to create structue of webpage or web app

-it is use to display data to user on browser

-it stand for Hyper Text Markup Language

-the latest version of html is html 5

-it is tag basis Language: <tag>

**In html we have two types of Tag:\***

1. paired Tag :<startTag> </endTag>

2.Unpaired Tag: <singletag>

**To create html file we have to write filename.html or filename.htm:**

step1: create html file with.html extension

step2: write code in it.

step3: execute it on browser

**<!DOCTYPE html>:** It tells search engine this is html type of document belong to version 5

**head :** it contain external links and meta data in it.

**body:** it is use to display information written in it.

**-Symantic element introduces:** the element with specific meaning.

-**New tag, new attribute and new Inputtype introduces** in html5.

-**Multimedia element introduces:**

**API introduces:**

-**Geolocation:** it return user Current position.

-**drag and drop :** grabbing an object and drag to another place

**-web worker :** it is small JavaScript code which execute at backend of webpage without affecting its current execution.

-**web storage**:it is use to store user data on browser.

**1.local storage**: it store user data on browser for permanently.

**2.session storage:** it store user data on browser for limited time.

-**SSE(Server Sent Event):** it allows use to get automatic update from server.

**elements in html**

<startTag attribute="value"> content </endTag>

**\*attribute : -**it gives additional information about tag.

-it always write in start tag.

-it comes with key value paired

**marquee Tag**: it will make scrolling content

**scrollamount:** it decide speed to scrolling content

**direction:** left, right, up, down

**behavior :** scroll, alternate, slide

**In html we have two types of element:**

**1. Block element:**

-it takes 100% width of its screen

-it always start on new line

-example : h1-h6,p,li,table,form

2**. Inline element:**

-it takes required width of its

-it never start on new line.

-example: b, i, u, a, img

**List:**

It is collection of items.

In html5 we have three types of list.

**1.Order List:** types =number, A, B, C, a, b, c, I, II, i, ii

**2.Unodered List:** type = disc, circle, square, none

**3.Description List:**

dl: Description List

dt: define Term

dd: define description

**How to insert images in webpage :**

-insert image by using relative path

-How to insert video file

-how to display audio file

**iframe :**

-It is use to generate small window in webpage.

-it has many uses

1.to display google map:

step1: go to google map and search location

step2: click on share => click on embed a map => copy html path

step3: paste path where you want.

2.to display YouTube video

3.to display one website into another.

**Table :** it is collection of row and column

in table first we create row and then column

**Q. What is difference between html and html5:**

Ans:

1. New Tag, Attributes, New Input type introduces: date, time, color, week, search, URL & New tags are not introduced in HTML.
2. Multimedia elements introduces in HTML5 & not introduced in HTML.
3. API introduces in html 5 & not introduced in HTML.
4. Semantic element used in HTML5 & not used in HTML.

**Q.what are the features of html5/: Symantic elements are used.**

New Tag, Attributes, New Input type introduces: date time color week search URL.

Multimedia elements introduces.

API introduces in html 5.

semantic element used.

Div & span are non-semantic element div is Block and span is inline .

**CSS Self Prepared Notes**

**(10/04/2024)**

**What Is CSS:**

used to design web application.

CSS Stands for Cascading Style sheet.

The latest version of CSS is CSS3.

**What are the ways to use CSS:**

1. **Inline CSS:** In starting tag we use style attribute in that we use property and value.

2. **Internal CSS:** In html page we use style tag inside we write CSS code.

**Syntax:**

selector{

property:value;

property:value;

property:value;

}

3. **External CSS:** In this we create a separate CSS file and that CSS file we import in html file head part by using link tag. Because of this html and CSS code will be in separate file. It will be easy to maintain.

**Selectors in CSS:** Used to Selecting an Element.

1. Element Selector: We use Element Name directly in Element selector.

2. Group Selector: We make group of elements and apply common css properties to them.

3. Id Selector(#): When you want to apply CSS Property to single element.

4. Class Selector(.): When we have to apply common bunch of CSS property to an multiple element.

5. Universal Selector: When we have to apply CSS property to all element available in Webpage.

**How to use Inspect Element:**

**BoxModel Concept:**

1. content: It can be any text or image
2. padding: It is used to clear area between content & element border.

3.border: A line drawn around an element.

4.margine: Used to clear area around element or Clear area in between element & screen window).

**Position Property:**

It is used to move element after applying Left, Right, Top, Bottom values. By default, every element has Property:

1. **Position: static;**
2. **Position: relative;** // Element move to Left, Right, Top, Bottom direction.
3. **Position: absolute;** // When we want child element must work according to it's parent element. For this parent element must have position: relative and child element must have position value.
4. **Position: fixed;**  //It is used to fix element on screen window.
5. **Position: sticky;**  //If you wants to stick an element to an border of window.
6. **Index:**

\*It is used to decide stack priority of an element. It has two values 1 and -1.

\*1 means heigh and -1 means low.

**Combinators:** Its tells relation between selectors.

1. descendant combinators(space)

2. child combinators(>)

3. adjacent sibling combinators(+)

4. general sibling combinators(~)

**Overflow:** It is used to fit big content into small BoxModel.

overflow:vissible

overflow:hidden

overflow:auto

**Opacity:** It is used to make background Transparent, Its value vary in between 0 & 1 only.

**\*Display:block;**  //when we have to convert inline element into block element.

**\*Display:Inline;**  //when we have to convert block element into inline element.

**\*Display:Inline-block;**  //When we have to apply height and width to an inline element in that case we use this property and value.

**Box-Sizing: Border-box:** It include element border size and padding size in its actual width.

**Media Query:**

It is used to apply different style rule according to the different screen width.

screen width >=1200px //Extra large width

screen width >992px //large width

screen width >768px //Medium width

screen width >=576px //small width

screen width < 576px //Extra small m width

**Syntax:** @media screen and (breakpoint)

{

CSS code

}

**Breakpoint:** CSS Rule will be applicable from

1.Max-length: 0px to It's Maximum Width.

2.Min-length: Minimum Width to Above.

**Display: Flex:**

-Used to manage the element layout on web page properly according to different screen width.

-we must apply display: flex property to parent element.

**Block element:**

These elements typically start on a new line and take up the full width available to them, pushing subsequent elements to a new line.

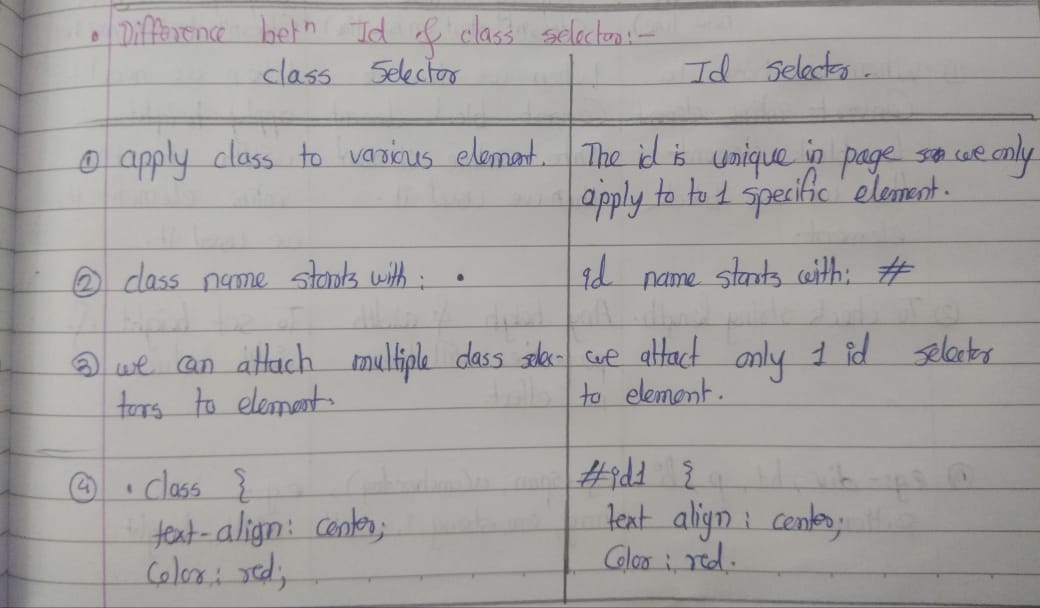
Example: <div>,<p>,<h1>to<h6>,<ul>,<ol>,<li>,<table>,<Header>,<footer>,<section>,<article>

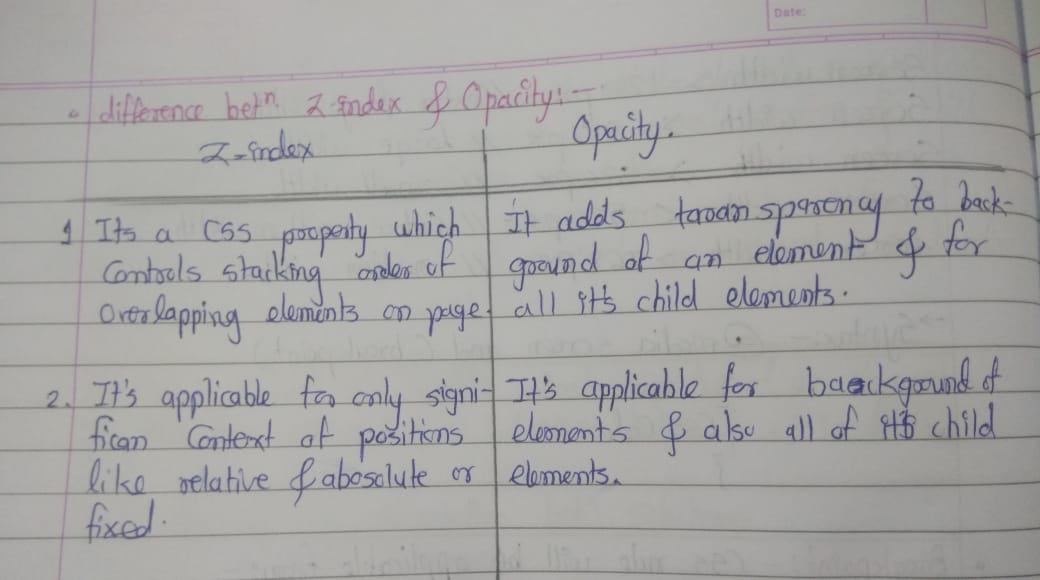
<blockquote>

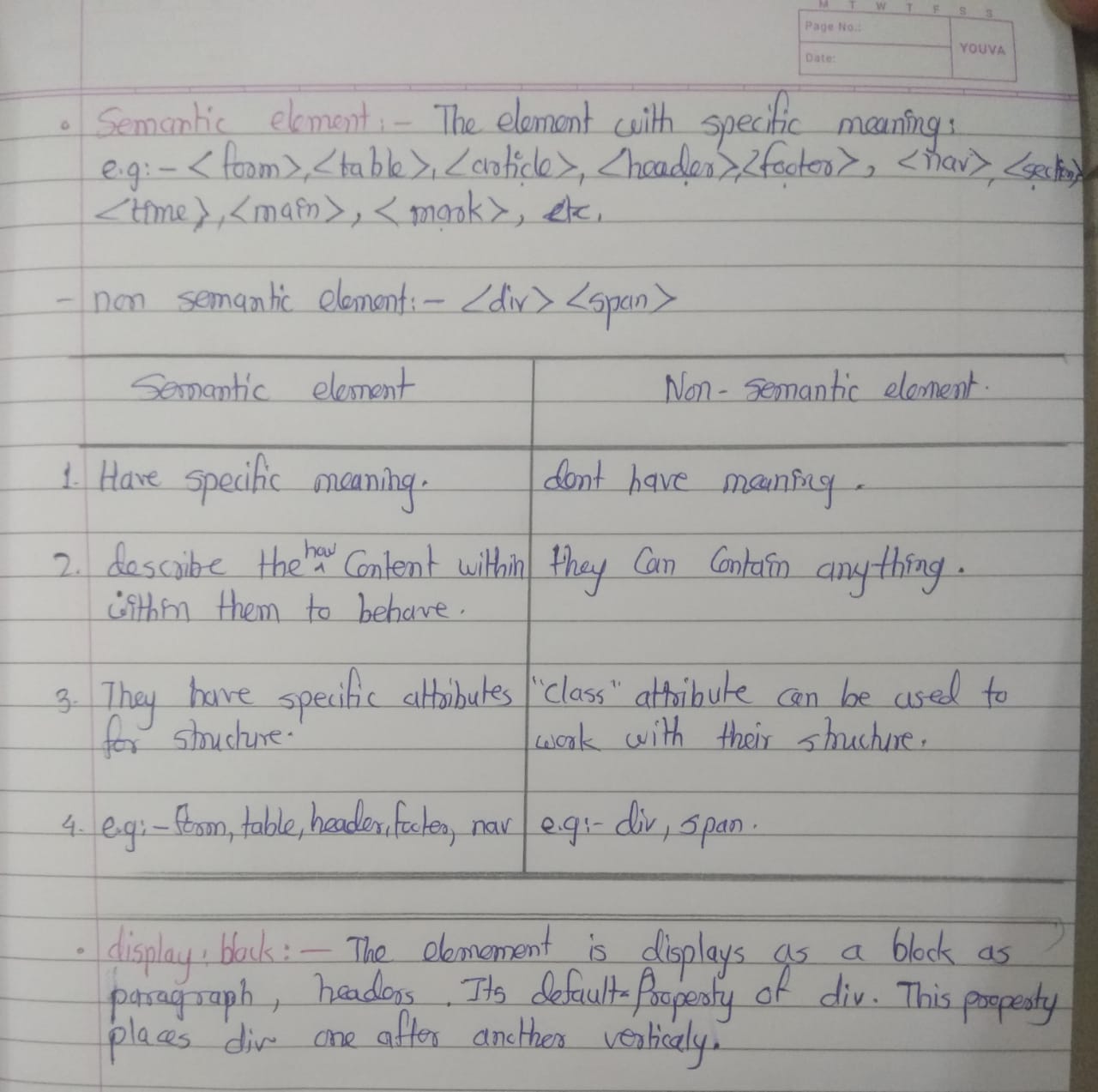
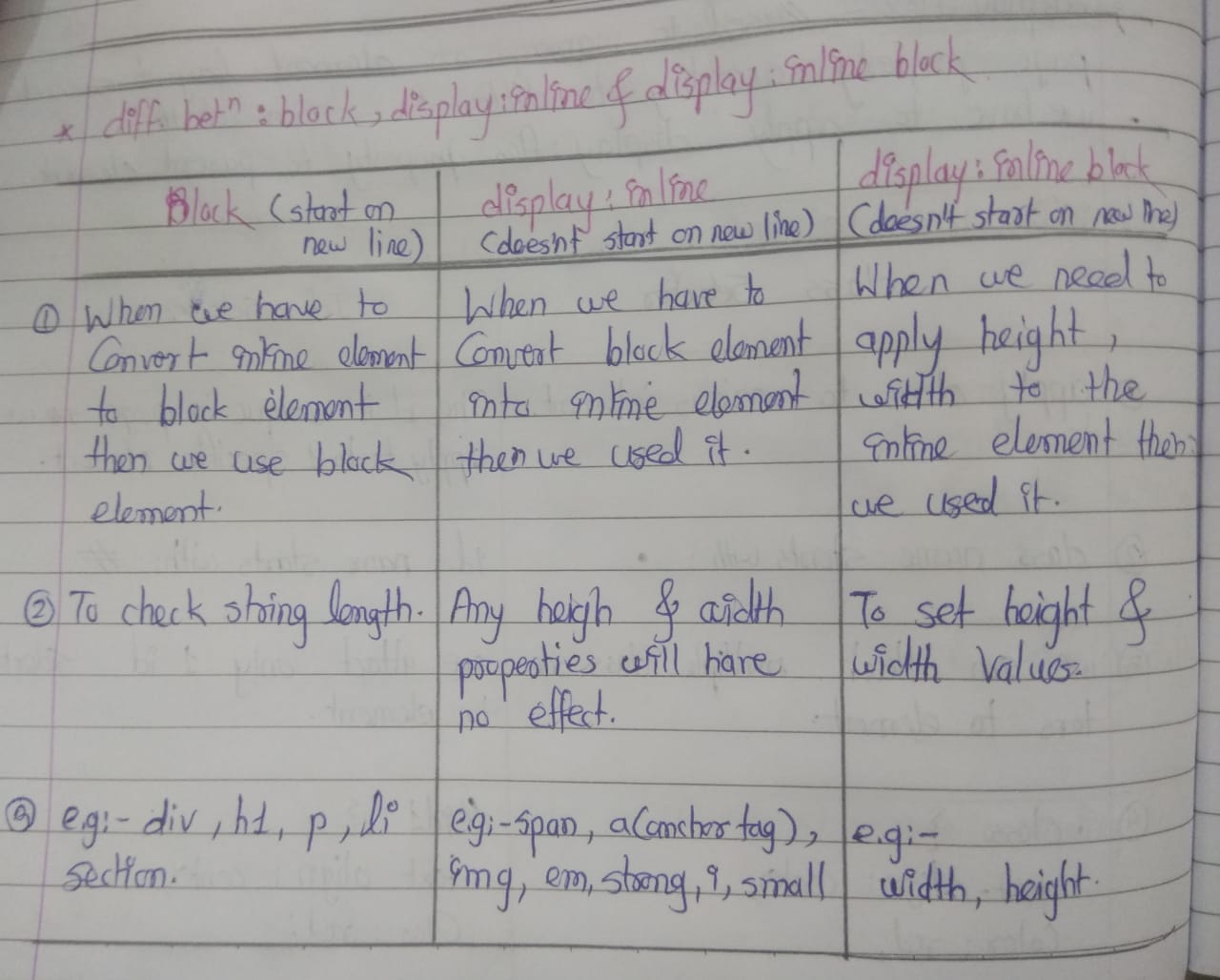
**Inline element:**

These elements do not start on a new line and only take up as much width as necessary, allowing other elements to sit beside them horizontally.

Example: <span>, <a>, <strong>, <em>, <img>, <input>, <label>, <br>







**Bootstrap Self Prepared Notes**

**Bootstrap Concept:**

* It is HTML, CSS, JAVASCRIPT framework.
* It provides In-built classes.
* The latest version of Bootstrap is Bootstrap5.
* To make the web application Responsive, Attractive & Mobile Friendly.
* Framework is collection of Multiple libraries, Library perform dedicated work.

**Types:**

1. By using Content Delivery Network (CDN) Link.
2. By Downloading.

**Bootstrap In-Built classes:**

.display-1 to .display-6

**.container class:**

It will take 15px margin from left and right side of an element as well as It convert Element into Block.

**Grid:**

Its is used to make webapp responsive.

In bootstrap grid it has two classes:

1. Row
2. Col

In grid row we can create columns with size 12 only. If Column size increase than 12 then that column will come on next row.

**Card:**

It is a container which includes products image with its information.

-Note: Sometimes In card only image is included.

**Form Validation in Bootstrap:**

There are two classes to perform form validation.

1. **.was-validated:** It will perform validation before submitting form.
2. **.needs-validation:** It will perform form validation after submitting form(For this you need to write JS code)

**To display message:**

**.valid-feedback:** It will work when no error.

**.invalid-feedback:** It will work when there was an error.

**Notes:**

* React & Angular both are Conceptualy same.
* **Ctrl+/ :** Used to Comments in html.
* **<hr> tag in html is unpaired tag**: to break statement and go to new line.
* **Its Official Website is:** getbootstrap.com
* **UTF Fullform:** Unicode transformation Format: Used to convert any data in bit.
* **Viewport:** Users visible area of a webpage.
* **Alt+shift+download arrow key**: copy same content to next line

**Javascript Self Prepared Notes**

**Javascript:**

* It is used to add functionality and behaviour into web app.
* It is loosly typed language.
* It is interpreted language.
* Javascript is object base language.
* In Javascript window is Global Object.

**There are Two ways to Use Javascript:**

1. Internal Javascript: In html file we use Script tag and we write Javascript code.
2. External Javascript: In this we create separate JS file and that JS file will import in html file by using script tag.

**How to display output in JavaScript:**

1. Document.write(): It will display output on webpage.
2. Window.alert(): It will generate one pop-up window to display output & It execute.
3. Console.log(): It will display output in browser console or terminal(used for testing purpose)
4. DOM(Document Object Model):

* When we create a variable in JavaScript, we cannot determine what type of data it contain until and unless initialize it.
* **let, var and Const** are three datatype variables which are used for to Initilize the value to the variable.
* == //It checks value only if both are same then it returns true result.
* === //It will check both Value as well as Its Datatype, if both are same then it returns true.

**How to create a variable in JS:**

**Syntax:**

let variable\_Name; // Variable declaration

variable\_Name=value; //Variable Initilization

let variable\_Name=value; //Defining a variable

**There are mainly two datatypes in JS:**

1. Primitive datatype: number,string,Boolean,undefine,null;
2. Non-Primitive datatype: array,object;

**Null datatype:**

To prevent to store a garbage value in variable for that purpose we use Null Datatype.

**Operators in Javascript:**

**Exponentiation(\*\*) :** It return power of any number.

e.g: 5\*\*3

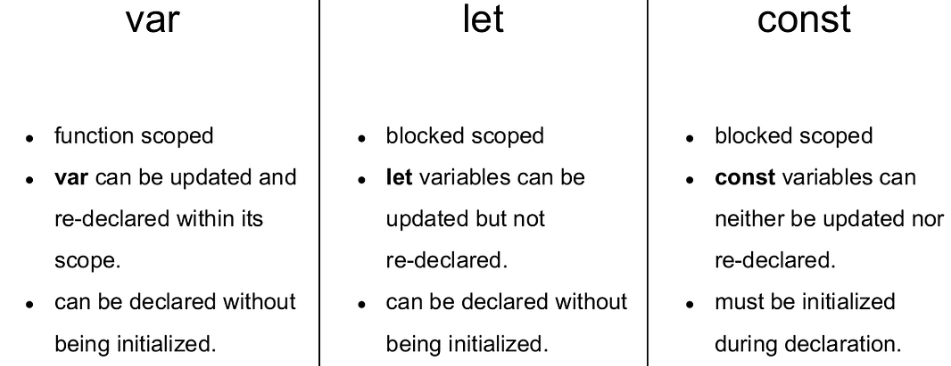
/

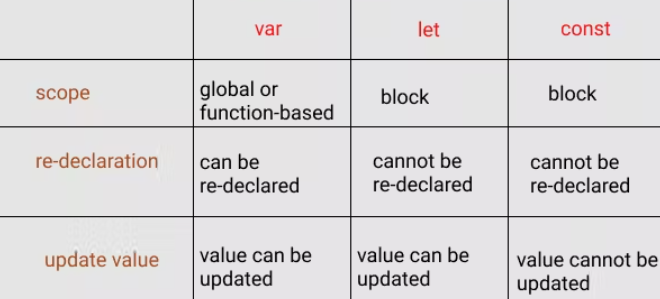
**Questions:**

1. **Why Javascript is called as Loosely Typed Language:**

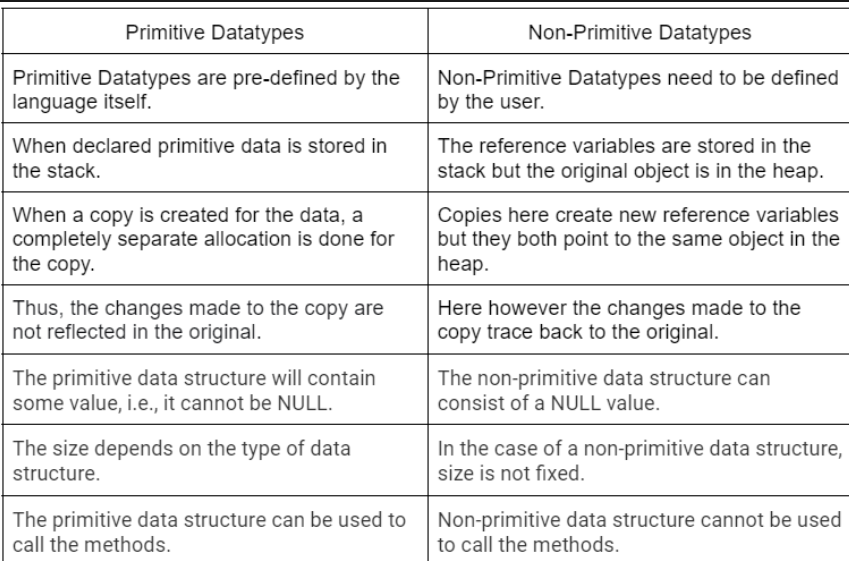
* A loosely typed language, means we do not have to specify what type of information will be stored in a variable.
* JavaScript automatically types a variable based on what kind of information you assign to it (e.g., that '' or " " to indicate string values).
* Many other languages, like Java, require you to declare a variable’s type, such as int, float, boolean, or String.

1. **What is Difference between let, var and const keyword?**

****

****

1. **Primitive & Non-Primitive Datatype Difference with example?**

****

**Primitive:** Strong, number, Boolean, undefined, null.

**Non-Primitive:** Object, array.

1. **Looping & Conditional Statements?**

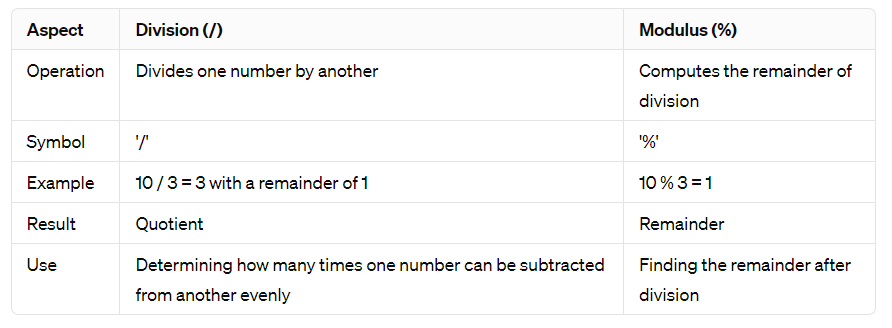
Looping Statements:

1. While
2. Do while
3. for

Conditional Statements:

1. if
2. if-else
3. else-if-ladder
4. nested-if-else
5. switch

1. **Difference Between Modulus and Division?**



1. **What is Difference between == & ===**

The == operator will compare for equality after doing any necessary type conversions. The === operator will not do the conversion, so if two values are not the same type === will simply return false. Both are equally quick.

**Function:**

* It is a block of code which will used for particular task or operation.
* Function will be executed only when we call it.
* It can be reusing multiple time.

**Syntax:** function functionName()

{

Code

}

**Types:**

1. In-built Functions:
2. User-defined functions:

**Anonymous Function():** A function with no name is called as “Anonymous function”

**Arrow Function():** Function with no name & No function keyword.

**Optional Parameterized Function():** In this parameter will optional not mandatory to pass value to it. always write at last.

**Rest Parameterized Function:** When we don’t know how many parameters are required in that case we use it.Here we use spread Operator.

**Array:**

It is used to store multiple values with same datatype.

In array values store on Index basis.

Array index always start from 0 index.

To access data outside array for that we write array[index\_number]

Array is Collection of similar type of data.

Array Stores its elements Sequentially, Linearly & Contagiously.

Array index is always start with 0 & Ends with size-1.

Array is denoted by []: Square bracket.

**How to use External Javascript:**

1. By using DOM how to set value.
2. By using DOM how to get value.
3. By using DOM how to apply style rule.

* **Push():** Add values at last index of Array.
* **Pop():** Remove values from last index of an array.
* **Unshift():** Add values at Starting index of an array.
* **Shift():** Remove values at starting index of array.

**Array methods:**

1. For of
2. For in

**Object:**

* It store multiple values with different datatype.
* In object data store in key and value pairs format.
* Denoted by: {}
* We can access object data outside by writing object.keyname

**Data stored in {} format**: It stores data in form of objects.

**How to insert Date & time in Webpage:**

-If you want to get date and time in javascript for that first we have to create object of date class.

-Syntax:

Let dateObj=new Date();

-Date class returns data f3rom browser.

**String & Its method:**

It is a collection of character.

In string every letter has index value and Its starts from index start from 0.

In string white space is also considered as string.

**In JavaScript we have 3 methods which return index of searching string:**

1. **indexOf (searching-string, start-index):**

* It return index of first occurrences of searching string.
* It search string from left to right.
* If searching string is not found then it return -1 as a result.

1. **lastIndexOf (searching-string, start-index):**

* It return index of first occurrences of searching string.
* It search string from left to right.
* If searching string is not found then it return -1 as a result.

1. **search(searching-string):**

* It return index of first occurrences of searching string.
* It search string from left to right.
* If searching string is not found then it return -1 as a result & It is faster than indexOf() method.

**There are 3 Methods which are return part of string by passing index number:**

1. **Splice(startIndex,endIndex):**

It will return part of string.

If we write +ve index then it will reach from left to right.

If we write -ve index then it will reach from right to left.

1. **substring(startIndex,endIndex):**

It will return part of string.

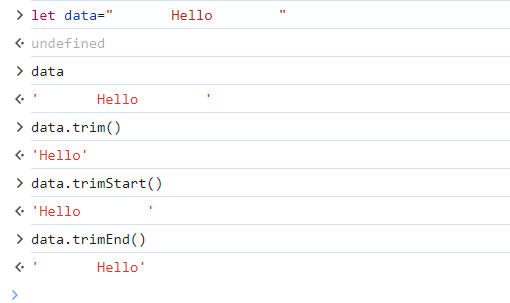
If we write +ve index then it will reach from left to right.

1. **subStr(startIndex,total-number-of-character):**

It will return part of string.

If we write +ve index then it will reach from left to right.

1. **Replace:**
2. **Trim() :** It remove unwanted space of both side string.



**\*callback() Method:**

Passing one function as an argument inside another function is called as “callback() method”.

When we write one function as an argument inside another function we do not use parenthesis.

**2 Methods:**

1. **setTimeOut(callback function,duration):**

It calls another method only once after a perticular time interval

1. **setTimeInterval(callback function,duration)**

It calls another method multipleafter perticular time interval.

**In 2009 (ECMA script-5 or ES-5**);

**In 2015 (ECMA script-6 or ES-6);**

-Where ECMA is stands for: **European computer manufacturers Association.**

**Que. How will convert Object type of data into JSON?**

In many programming languages, you can convert an object type of data into JSON format using built-in functions.

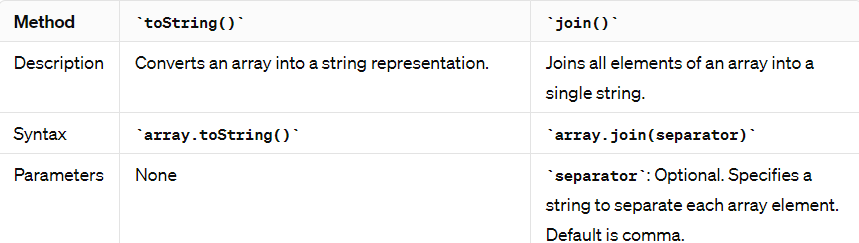
**Syntax:**

Let obj={name=”Pranu” age:60,city:”Pune”};

Let json=JSON.stringify(obj);

Console.log(json);

**Que.Difference between toString() & Join() method:**

****

**Notes:**

* We already know what type of data stored in code is called: Strict Type Language
* C & CPP are the Compiled Language, and It is Loosely Coupled Language.

**Que:**

* **For Each Method:**

**Ans:** The forEach() method calls a function for each element in an array. It is not executed for empty elements.

## **Syntax:**

array.forEach(function(currentValue, index, arr), thisValue)

**Example:**

let sum = 0;  
const numbers = [65, 44, 12, 4];  
numbers.forEach(myFunction);  
  
function myFunction(item) {  
  sum += item;  
}

* **Map method:** To performing a manipulation.

**Filter method:**

Filter() method creates a new array filled with elements that pass a test provided by a function.

The filter() method does not execute the function for empty elements.

The filter() method does not change the original array.

**…………………………………………………………………………………………………………………………**

**13/04/2024**

**OOP’s Concepts:**

-OOP is stands for Object Oriented Programming.

* It provides Security to your code.
* Code Reusability.
* Easy to maintain.
* Robust.
* High level Programming Language.

**Que. What are the Features/Fundamentals of OOP’s Concept:**

1. **Class:**

-It is Collection of data members & member function.

-It is template (It means a design which we can reuse) which we can reuse.

-To create a class, we use class keyword.

-Class name must start with Capital letter (we create function, variable from lower case so it specify different than others we use it as capital).

1. **Object:**

-It is an instance of class.

-With the help of object, we can access class data members & member functions outside class.

**Syntax:**

let objName=new Class\_Name;

**Constructor:**

* Constructor is a special member of a class which execute automatically when class object is create or initialize.
* Constructor name must only constructor in JavaScript.

1. **Inheritance:**

-Accessing properties from parent class to child class.

-We use extends keywords in between two class names.

**Use of Super ():**

It represents class constructor inside child class constructor.

It must be first statement in child class constructor.

1. **Encapsulation:**

Wrapping a data into single unit.

1. **Abstraction:**

Hiding unnecessary data from user and showing only required thing. (e.g: ATM Background details hidden from Users)

* **Static data member & member function** will be accessible outside class **with class name only.**
* To **make static data member** we use **“static” keyword.**

1. **Polymorphism:**

Its ability to take More than One Form.

1. **Method Overloading:** We have multiple method with same name in different class (e.g. One home with Twins)
2. **Method Overriding:** When we have multiple method with same name in same class.

**Module Concept:**

-It contain single or multiple JavaScript file which has specific use. In this If you want to share data from one file to another for that purpose we use export keyword.

-If you want to access data from another file to your file for that we use “import” keyword.

-When we need to use class member of that class inside that class body then we use this keyword.

**DOM [Document Object Model]:**

-It allows us to create, manipulate html element in JavaScript.

**Form Validation:**

-**Form validation** in JavaScript involves ensuring that user-submitted data meets certain criteria before it's sent to the server for processing. This helps maintain data integrity and prevents erroneous or malicious data from being submitted.

-**HTML Code:**

<form id="myForm">

<input type="text" id="username" placeholder="Username">

<input type="email" id="email" placeholder="Email">

<input type="password" id="password" placeholder="Password">

<button type="submit">Submit</button>

</form>

**JS Code:** Also required for Form Validation.

**Note:**

* When we create function in class, we don’t use function word in class because it is Loosely typed language.
* Ctrl + ` : Is used to open terminal
* Steps for OOPS code:
* Write a code:

Open terminal using ctrl + ` and type node Filename.js //OOPS.js Code File

* In java There is no any access specifiers present in JavaScript.
* There is no Interface present in JS.

Identifier means meaningful variable.

* “=” it is Assignment operator.

**Download and Installation difference:** when you download code, we can use it but when we install it then we use it easily.

**Difference Between Login and Sign in:**

* In Login, accessing an existing account by providing credentials (username/password) & In Sign in accessing an existing account by providing credentials (username/password).
* Login is specific to accessing an existing user account & Sign in can refer to accessing an existing account or creating a new one.

**………………………………………………………………………………………………………………………….**

**15/04/2024**

**JQuery Self Prepared Notes**

**(15/04/2024)**

**JQuery:**

It is JavaScript Library & It simplifies JavaScript code.

It is easy to learn.

It is open source Library.

The main purpose of JQuery is to make JavaScript code much easier to use it in application.

**What we can do with JQuery:**

1. DOM Manipulation
2. Use CSS
3. Use JavaScript Event.

**There are Two ways to use JQuery:**

1. By using CDN (Content Delivery Network) Link
2. By downloading.

**Official Website for JQuery:** jquery.com

**Syntax for using JQuery:**

$(document).ready(function(){

//logical code

})

-Here**, $ is Preprocessor** which Preprocesses the Code.

-**ready()** It prevents the JQuery code to execute before html or document is not loaded.

**Selectors:** It is used to select an element, It plays most important role in JQuery.

1. Element selector(): Here we use element name directly.
2. Id selector(#): It is used to select single element.
3. Class selector(.): It is used to select multiple elements.

\***This keyword:** it will represent current element.

* Here we saw event click, mouseover, mouseleave.

**$(".myName").mouseover(function (){**

**$(this).css({"color":"blue","text-transform":"uppercase"});**

**});**

**$(".myName").mouseleave(function (){**

**$(this).css({"color":"red","text-transform":"lowercase"});**

**});**

* We saw How to use CSS
* We saw How to use multiple event

**On method:** It is used to apply multiple events to an single element.

**Focus and Blur event:**

$(document).ready(function(){

    $("input").focus(function(){

        $(this).css({"border":"4px solid blue","border-radius":"5px","width":"100px","height":"50px","text-align":"center","background-color":"lightgray"})

    })

    $(document).blur(function(){

        $(this).css({"border":"4px solid blue","border-radius":"5px","width":"100px","height":"50px","text-align":"center","background-color":"lightgreen"})

    })

**FadeIn, FadeOut, FadeToggle:**

$("#btnHide").click(()=>{

$("img").fadeIn();

})

$("#btnShow").click(()=>{

$("img").fadeOut()

})

$("#btnToggle").click(()=>{

$("img").fadeToggle(2000);

})

**SlideIn, SlideUp, SlideToggle:**

**Animate:**

To apply first we must change element relative property only then we can apply “Animate” effect.

**…AJAX…**

**(15/4/2024)**

**What is AJAX?**

-AJAX is not a programming language.

-AJAX is a technique for accessing web servers from a web page.

-Asynchronous JavaScript & XML (Extensible Markup Language).

-AJAX allows you to send and receiving data asynchronously without reloading the web page, So it fast.

-AJAX is mostly used from client side.

**Where It Is Used:**

-There are too many web applications running on the web page that are using ajax technology.

-Like: gmail, facebook, twitter, google map, youtube etc.

**AJAX Just Uses a Combination of:**

-A browser Built-in **XML HttpRequest Object** (to request data from a web server).

-JavaScript and HTML DOM (to display or use the data).

-With AJAX we can update parts of a web page without reloading the whole page.

**The Keystone of AJAX is the XMLHttpRequest object:**

Create an XMLHttpRequest.

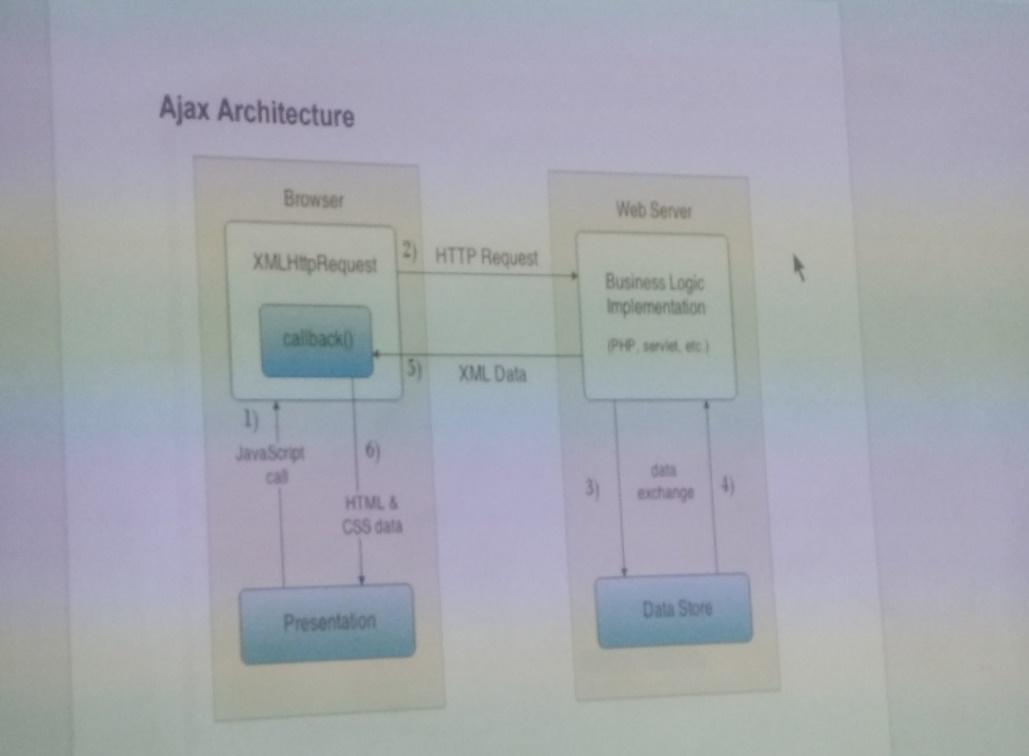
Define a callback function.

Open the callback function.

Send a request to server.

**AJAX Architecture:**

**Diagram:**



1. **Create an XMLHttpRequest Object:**

* The XMLHttpRequest object can be used to exchange data with a web server behind the scenes. This means that it is possible to update parts of a web page, without Reloading the web page.
* Example: variable\*new XMLHttpRequest ();

1. **Define a callback function:**

* In this case, the callback function should contain the code to execute when the response is ready.
* Example: xhttp onload=function {

//What to do when the response is ready

}

1. **Send a Request:**

* To send a request to a server, you can use the open () and send () of the XML HttpRequest object.
* Example: xhttp. open (“GET”,” ajax\_info.txt”);

Xhttp.send();

**AJAX Workflow:**

JSON: it is a JavaScript Object Notation.

It looks like a JS object, In that data store in a key and value pairs format.

Its denoted by,{} and key abd value are separated by:

If you want to access data outside it for that we write object, keyname.

Basically data is available in text format.

**JSON data is more convenient to fetch from server and send to server.**

**How to upload database on JSON-server:**step1: json-server ==watch -p 9999 db.json

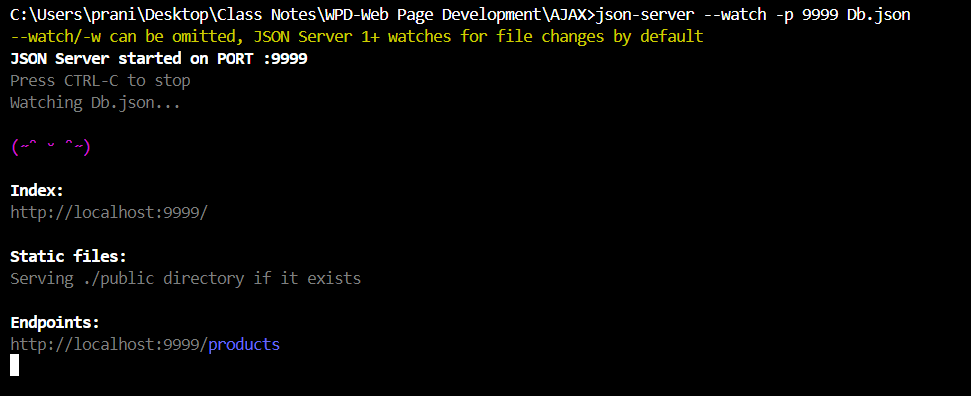
**\*how to intall json:**

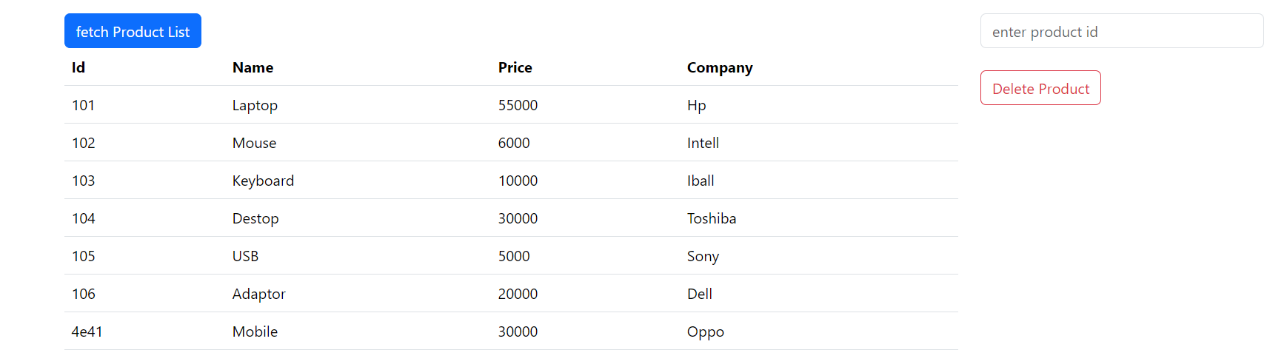
npm install -g json-server

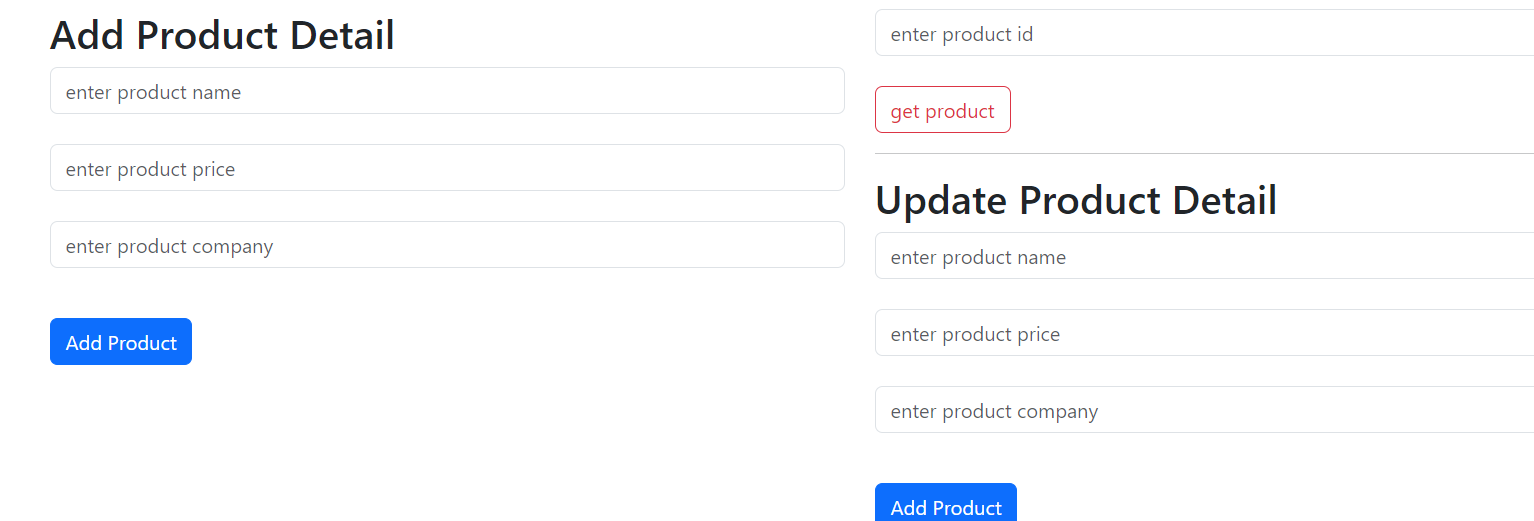
**To Run Product.list Code in your laptop:**

json-server --watch -p 9999 Db.json

Then Open Product.list code & Run it on chrome







**NodeJS Self Prepared Notes**

**(16/4/2024)**

**What is Node.JS:**

* It is an open source server side runtime environment built in chromes V8 JS engine.
* It provides an event driven, non—blocking(asynchronous) I/O and cross platform runtime environment for building highly scalable server-side application using JavaScript: Event ke through hone wale operations, Asynchronous operations & Provides Cross platform environment for build server side application using JS through the help of Node.JS.
* Node.JS can be used to build different types of application such as command line application, web Application, REST API server etc.
* It is mainly used for build Network programs like web servers, PHP, JAVA or ASP.Net.
* Node.JS was written & introduced by **Ryan Dahl in 2009.**

**Advantages:**

1. Nodes.js is an open-source framework.
2. Uses JS to build entire server side applications.
3. Lightweight framework that includes bare minimum modules.
4. Asynchronous by Default. So it performs faster than other frameworks.
5. Cross-platform framework that runs on windows, MAC, or Linux.

**Setup:**

1. Node.js
2. Node package Manager (NPM)
3. IDE (Integrated Development Environment) or Text Editor

**Node.JS Console REPL:**

* Node.js comes with virtual Environmemt called REPL (aka node shell)
* REPL stands for Read-Eval-Print-Loop.
* It is a quick and easy way to test simple Node.js/JS Code.
* To launch the REPL, open cmd and type node (press enter) // ctrl+d: bahar aane ke liye

**REPL e.g:** Arithmetic operation, string concatination operations.

**For Save Code File:**

.save filename. extension

**Check Node.js is Available or not in System:**

Node -v

Npm -v //It will display NPM version

**How to enter REPL Environment:**

To enter REPL Environment for that write node (press enter).

* Save code from REPL Environment to file.

.save filename.js

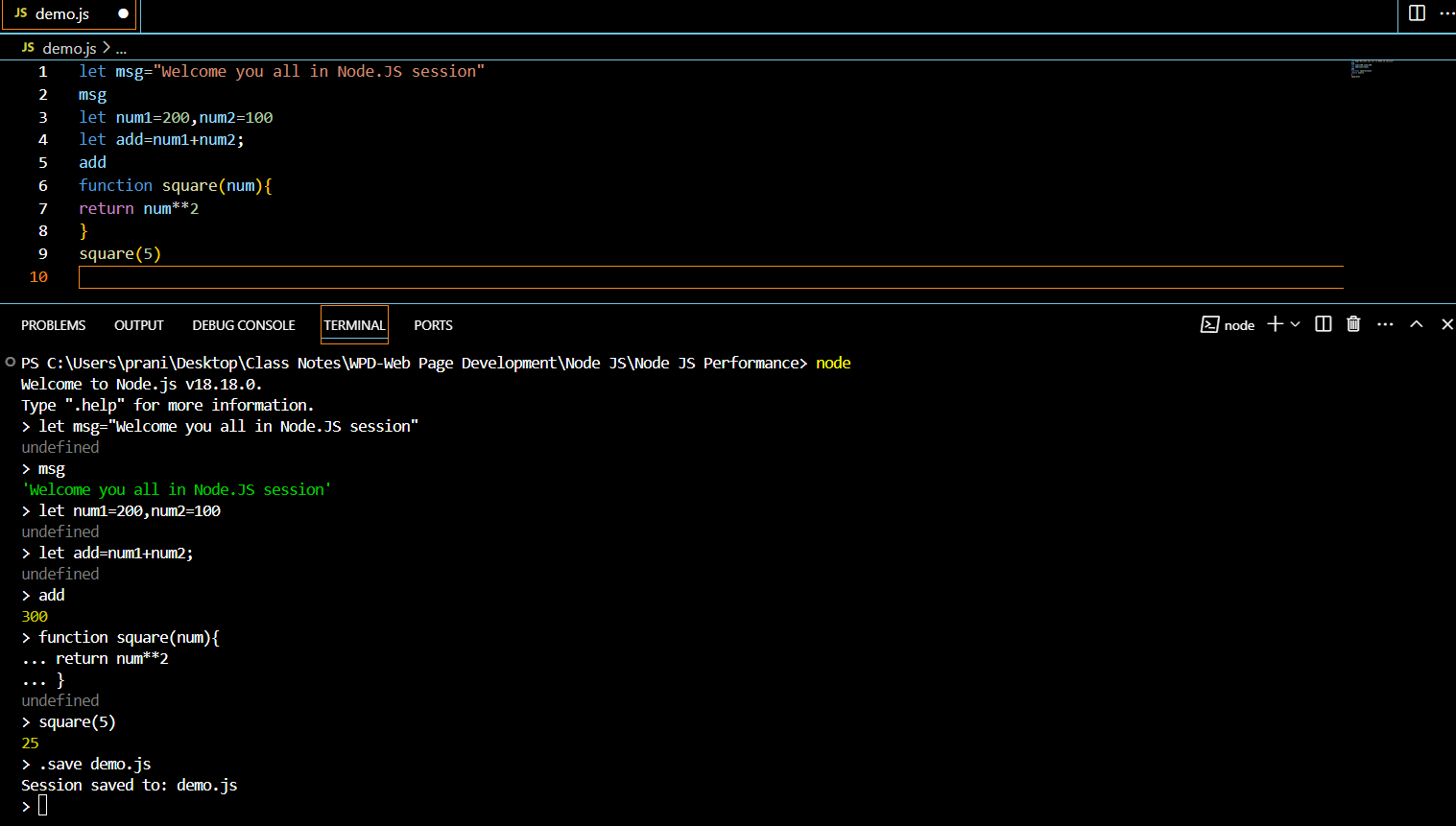
* To load code from file to REPL Environment.

.load filename.js

* To come out from REPL Environment // ctrl+d

**Demo Code:**

Run on Terminal and Reflects on Code file.





**Node.JS Module:**

* Module in Node.JS is simple or complex functionality organized in a single or multiple JS Files which can be reused throughout the Node.JS Application.
* Each module in Node.JS has its own context, so it cannot interfere with other modules.
* Each module can be placed in a separate.js file under a separate file.

**Node.JS Module Types:**

1. **Core Modules**: //Inbuilt Modules
2. **Local Modules:** //We created
3. **Third party Modules:** //use already Built & Installed modules.

**Core Modules:**

* The core module includes bare minimum functionalities of Node.js.
* These core modules are compiled into its binary distribution and load automatically when Node.JS process starts.
* However, we need to import the core module first to use it is in your application.

**Example of Core Module:**

1. **http:** To create Http Server in Node.JS
2. **os:** Provides information regards Operating System.
3. **path:** To handle the file path.
4. **url:** To parse the URL string.
5. **fs:** To handle the file system in computer.
6. **querystring:** To handle the URL query string.
7. **util:** To access Utility function.

**Node.JS Modules:**

1. **Core Module:** Module Provide by Node.JS to Us.
2. **Local Module:** Module created by Developer for Own Use.
3. **Third Party Module:** Module developed by Third Party Vender.

* First, we need to install it & then Use It & then Import it.

**Core Modules:**

1. **Path module:**

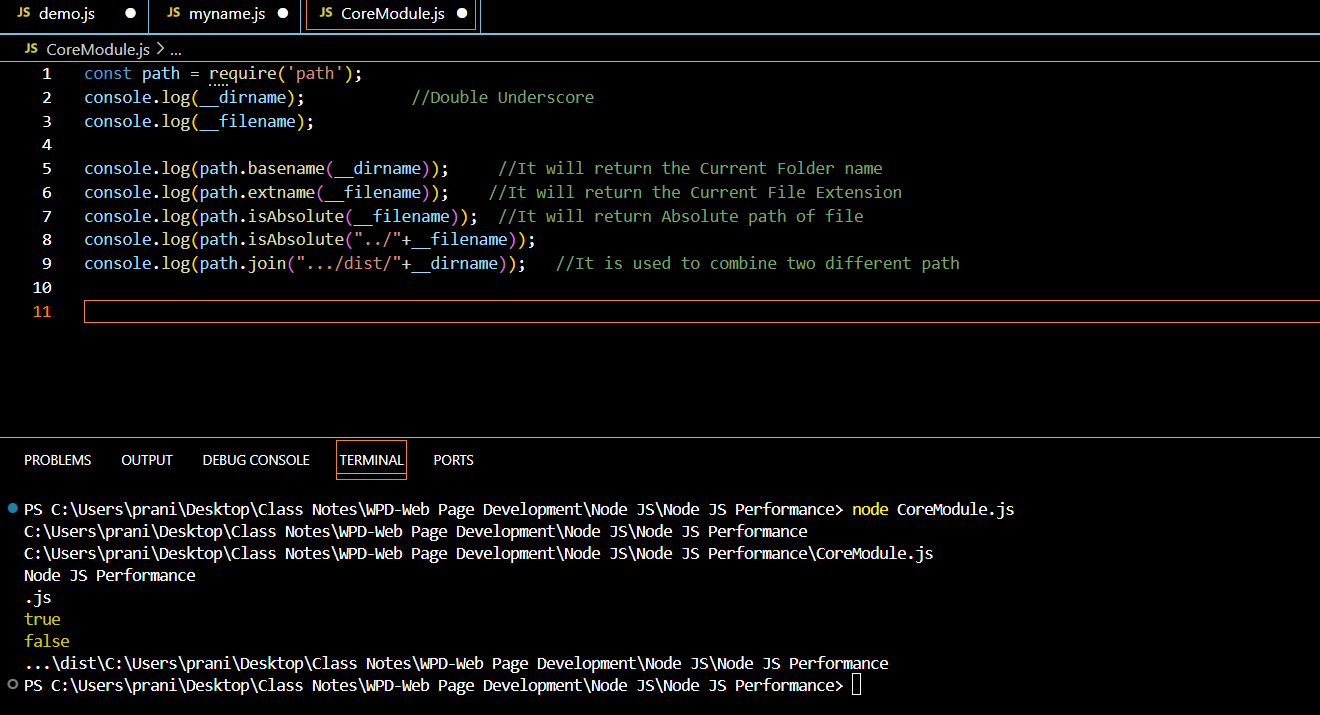
It provides a lot of very useful functionality to access & interact with the file system.

It is used for file related path operations.

In node.js by default we have Two Object:

\_**dirname:** It will return Current directory.

\_**filename:** It will return current filename with directory path.



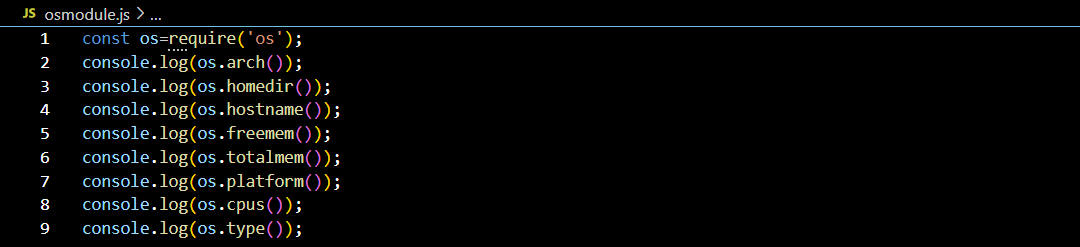
1. **OS Module:**

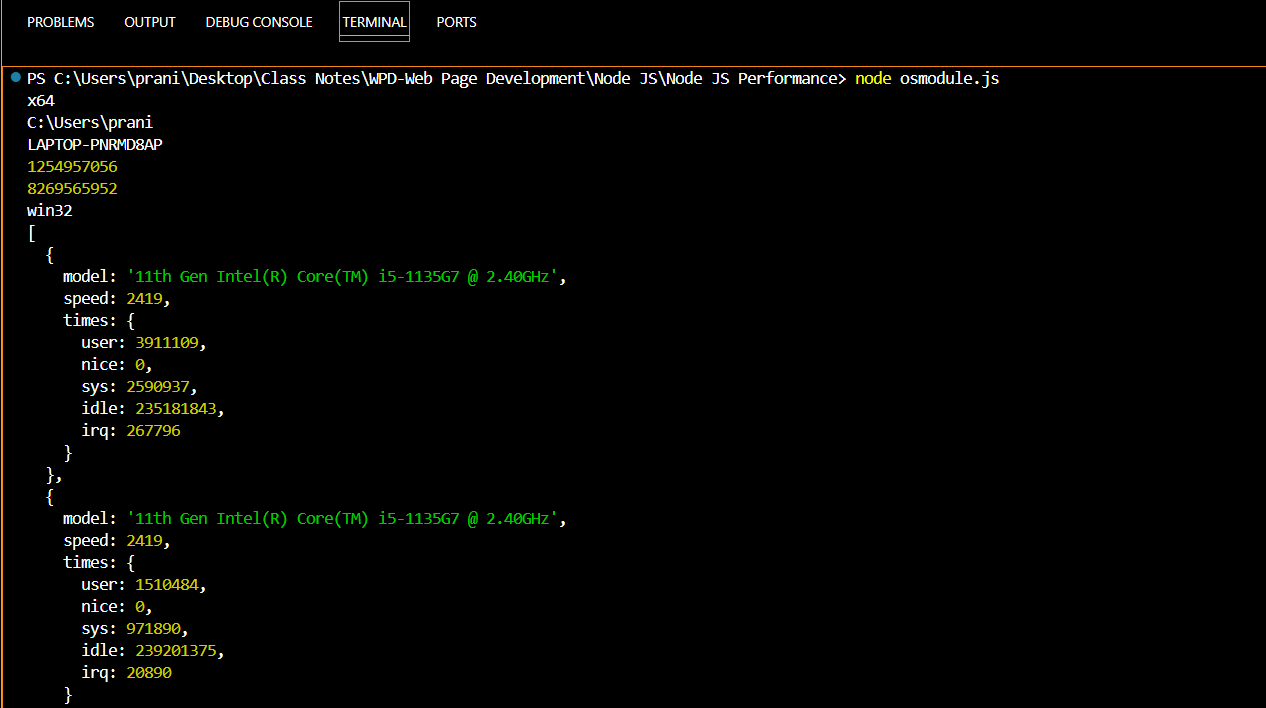
This module provides information of OS and the computer the program runs on, & interact with it.

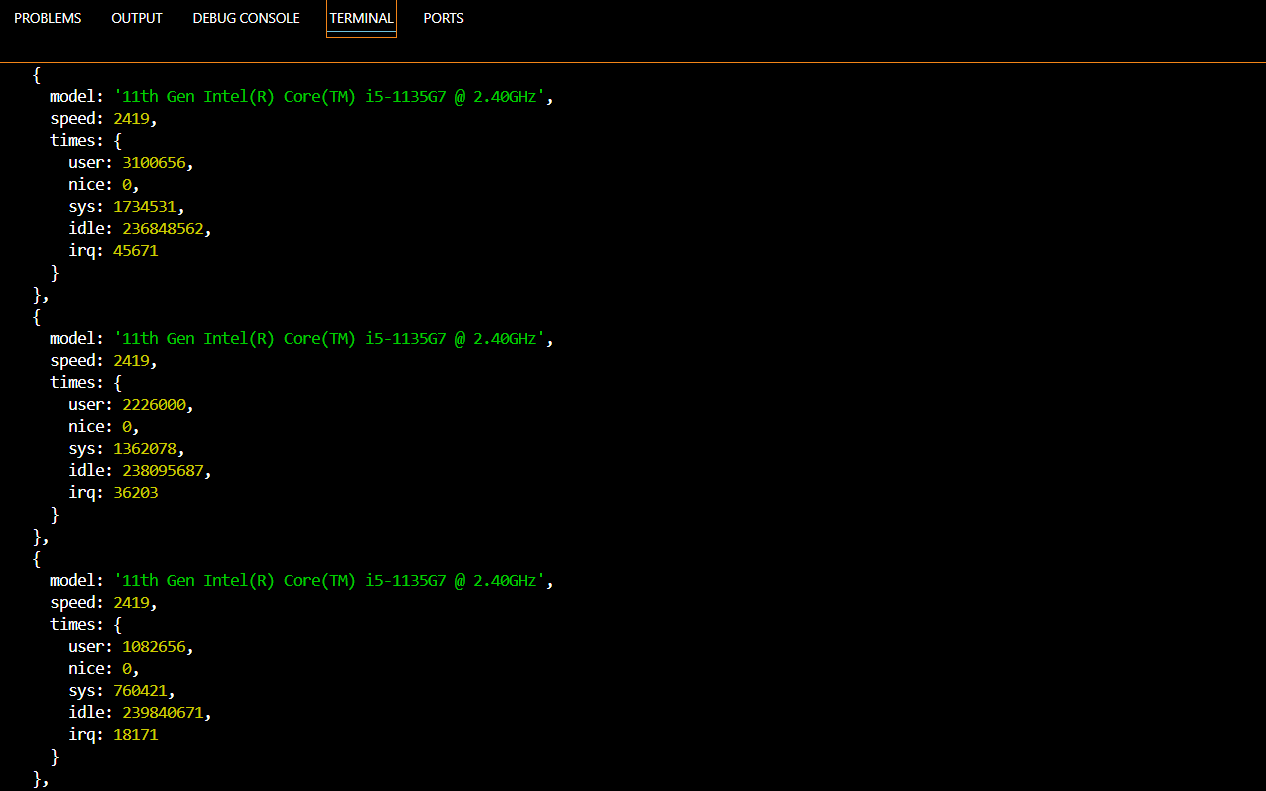
Const os= require(‘os’);

**Method:**

* **Os.arch():** Return the string that identifies the underlying architecture like arm,x64,arm64.
* **Os.freemem():** Return the number of bytes that represent the free memory in the system.
* **Os.homedir():** Return the path to the home directory of the current user.
* **Os.totalmem():**Returns the number of bytes that represent the total memory available in system.



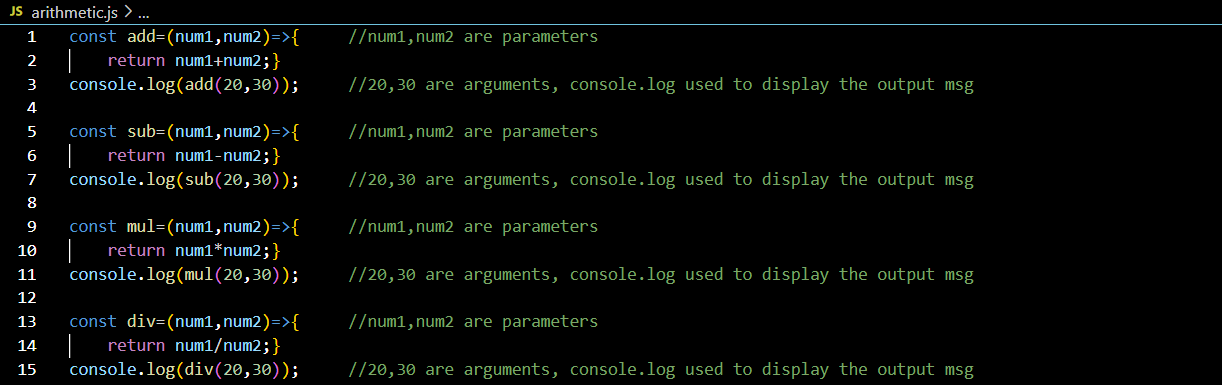


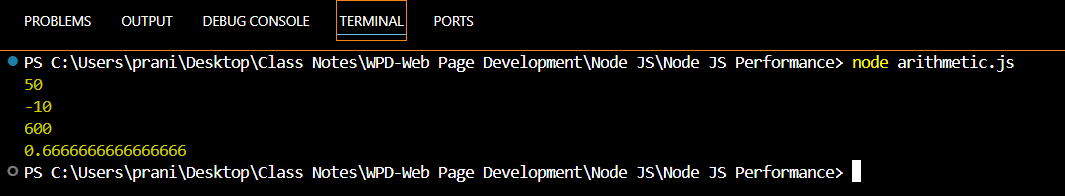




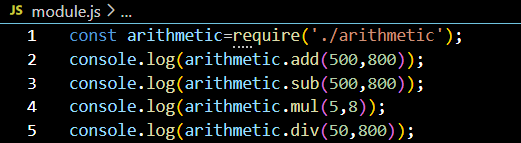
1. **Local Module:**

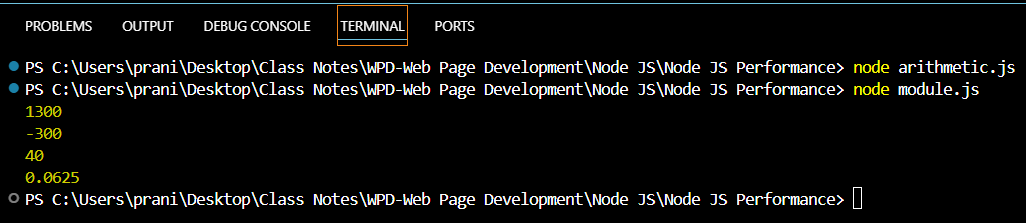
* The modules which are created locally in your Node.JS module.
* These modules include different functionalities of you application in separate files and folders.
* You can also package it and distribute it via NPM, So that Node.js community can use it.
* Export Literals
* Export function
* Export function as a class.
* The module.exports is a special object which is included in every JS File in the Node.JS Application by default.
* Use module.exports or exports to expose a function,object or variable as a module in Node.js.
* The require() function will return an object, function, property or any other JS type, depending on what the specified module returns.





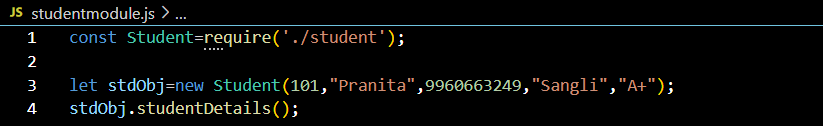
**Arithmetic operations code**

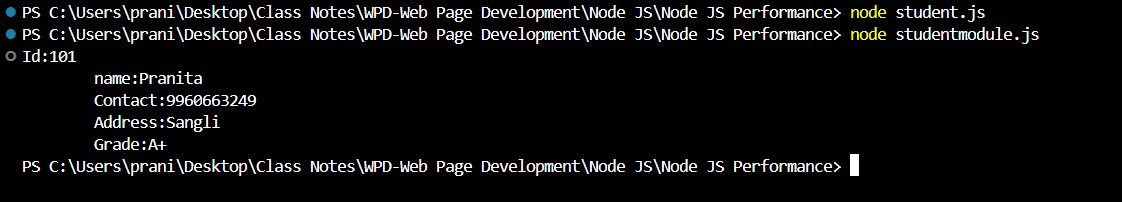




**Task: Create one class with name Student containing stdId, stdName, stdContact, stdAddress, stdGrade and one function with name studentDetails which will display student all information.**

****

****

****

1. **NPM: Node Package Manager:**

* Node Package Manager (NPM) is a command line tool that installs, updates, or uninstalls Node.js packages in your application.
* It is also an online repository for open-source Node.js packages.
* The node community around the world creates useful modules and publishes them as packages in this repository.
* NPM is included with Node.js installation and You can verify using npm-v command.
* If you have an older version of NPM then you can update it to the latest version using the following command:

Npm install npm -g.

**Command Use With NPM:**

Npm install module-Name –sav //Install third party module locally and add dependency in package.json file

Npm update module-Name //Update third party module

Npm update module-Name -g //Install third party module globally

Npm uninstall module-Name //uninstall third party module

**How to Use Third Party Module:**

* Step1: Install module using npm like=> **npm install module-name-save**
* Step2: Import it in file where you want=> **var chalk=require(‘chalk’)**

**Name of some third party module:**

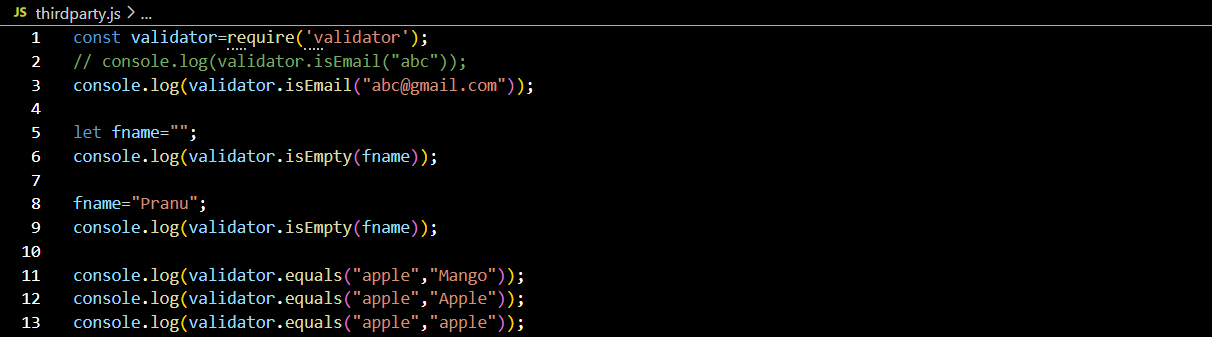
1. Chalk,
2. Validator
3. express

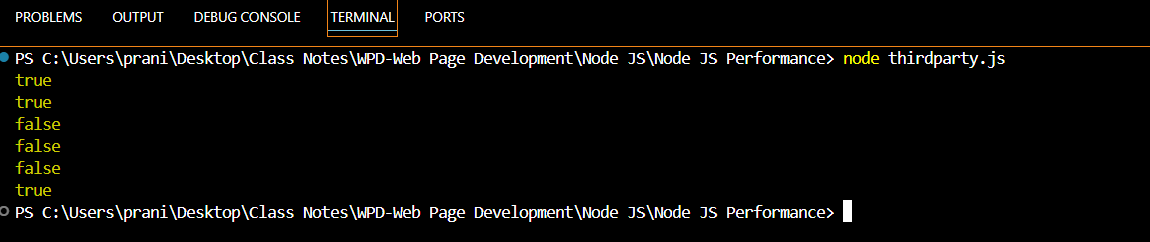
**Third party module:**

* First, we must install it.
* Then import it where want.
* After installation we have import it where we want.
* You will get all third party module and library in npmjs.com.

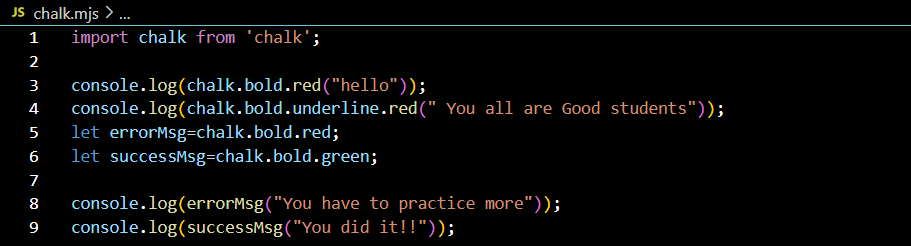
**Use of Validator module:**

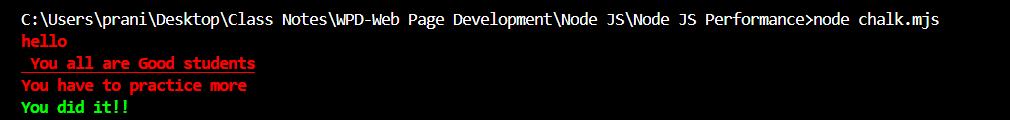
Npm i validator //In command prompt Run it





**Use of Chalk Module:**

****

****

**Blocking & non-blocking:**

* Blocking is when the execution of additional JS in the Node.JS process must wait until a non-Javascript operation completes.
* This Happens because the event loop is unable to continue running JS while a blocking operation is occurring.
* All of the I/O methods in Node.JS standard library provides asynchronous versions, which are non-blocking and accept callback functions. Some methods also have blocking counterparts, which have names that end with sync.
* Blocking methods execute synchronously and non-blocking methods execute asynchronously.

1. **Blocking: Synchronous**
2. **Non-Blocking: Asynchronous**

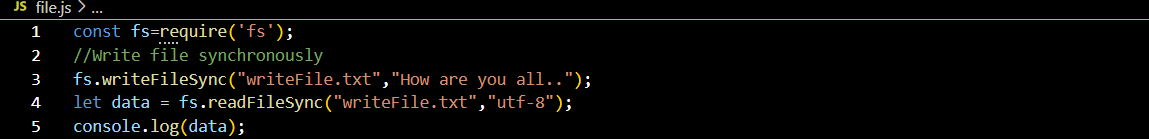
**Callback:**

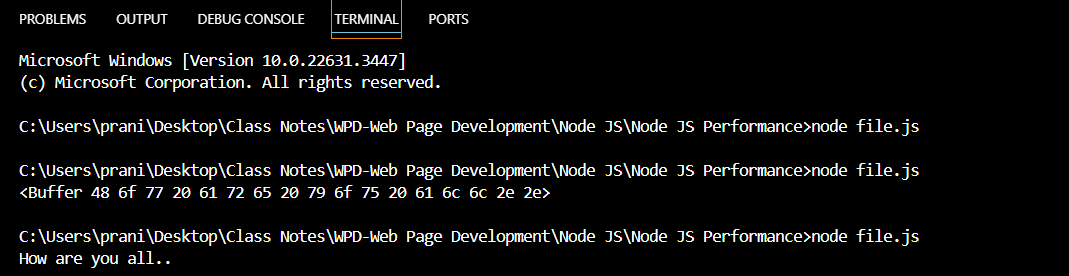
* A **Callback is a function called at the completion of a given task,** this prevents any blocking and allows other code to be run in the meantime.
* The general idea is that the callback is the last parameter. The callback gets called after the function is done with all its operations. Traditionally, the first parameter of the callback is the error value. If the function hits an error, then they typically call the callback with the first parameter being an error object. If it cleanly exists, then they will call the callback with the first parameter being null and the rest being the return values.

**Events:**

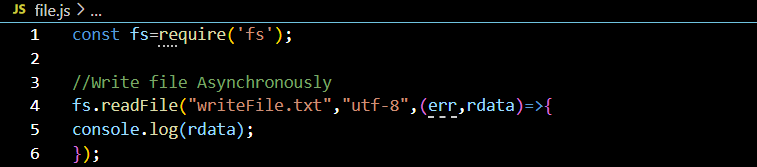
* Every action on a computer is an event,Like when a connection is made or a file is opened.
* Node.JS has a built-in module,called “Events”, where you can create,fire and listen for your own events.
* All event properties and methods are in instance of an EventEnitter object.To be able to access these properties and methods,Create an EventEmitter object.
* // Get the reference of EventEmitter class of events module: Var events=require(‘events’);
* //Create an object of EventEmitter class by using above reference: var em=new events.EventEmitter();
* Emit is used to trigger an event.
* On is used to add acallback function that’s going to be executed when event is triggered.

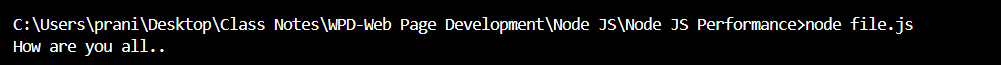
**For Synchronous file: Sync word before filename.**

****

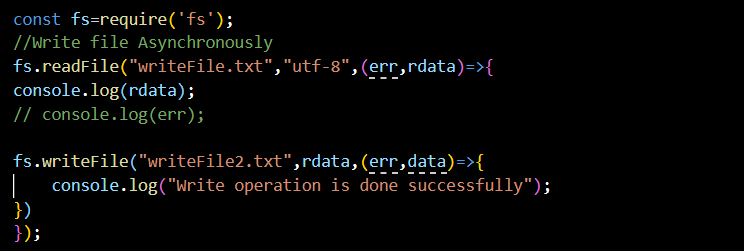
****

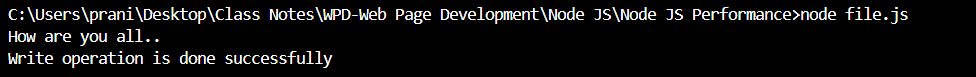
**For Asynchronous file: no any word before filename.**



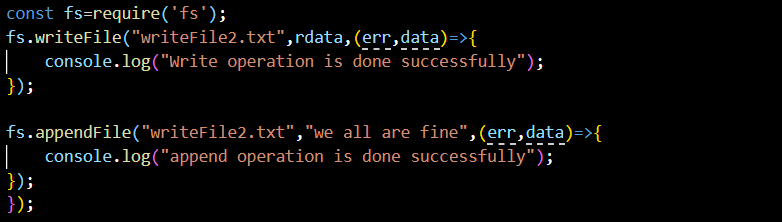


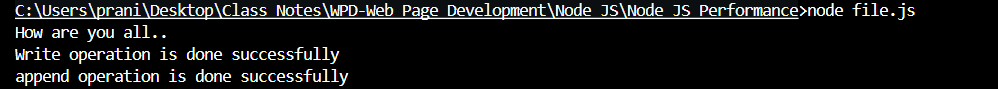
**Write asynchronously with error:**



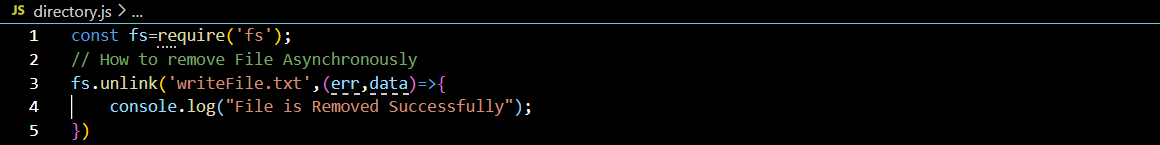


**Append the Data into file:**



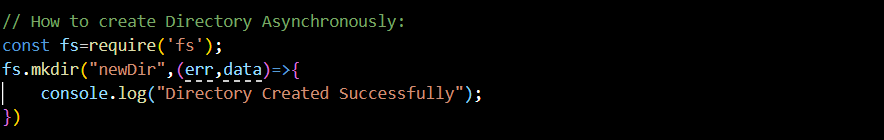


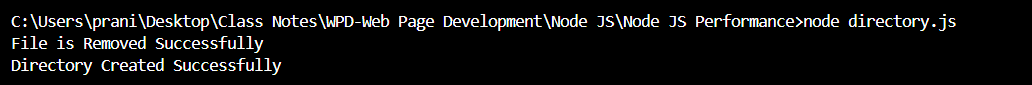
**How to Remove File Asynchronously:**



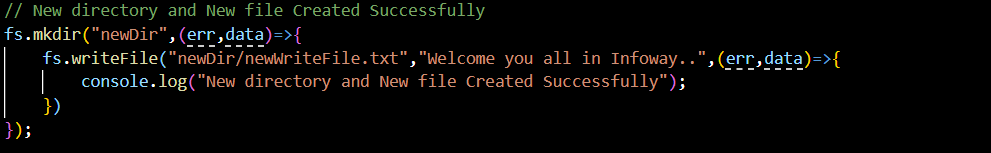


**How to create Directory Asynchronously:**

****

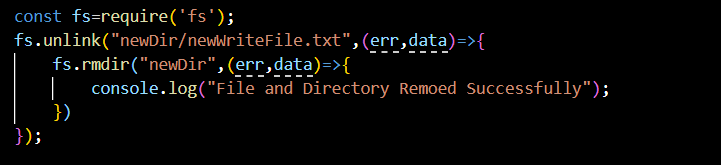
****

**New Directory & New File Created Successfully:**



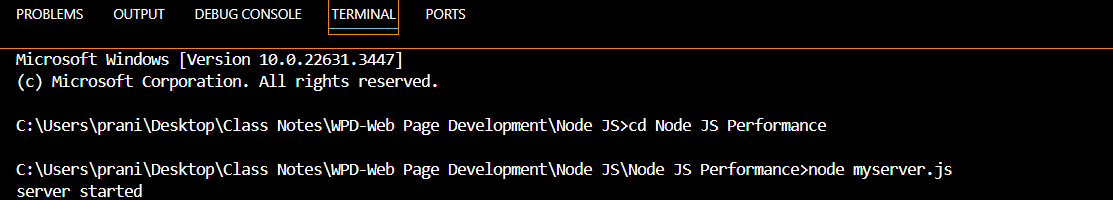


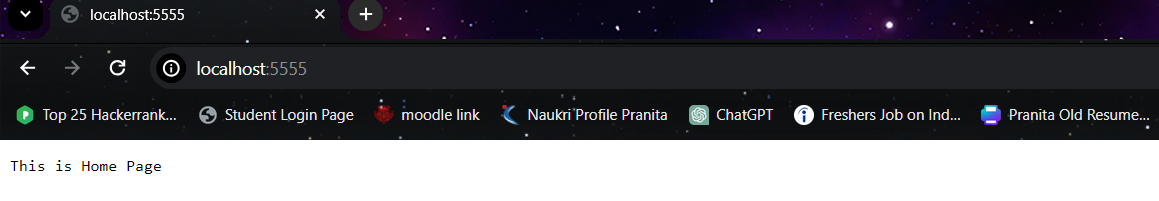
**New Directory & New File Removed Successfully:**

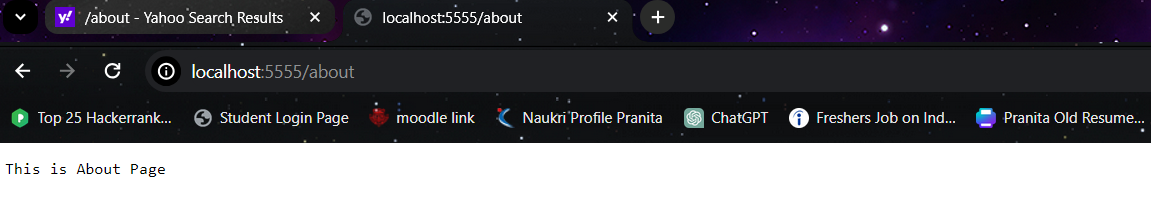
****

****

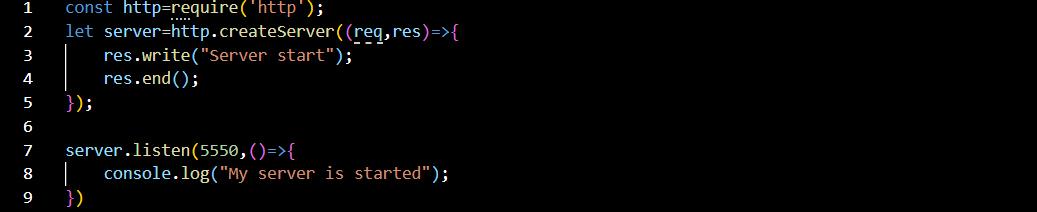
**On Server By default Index file is opened in server:**

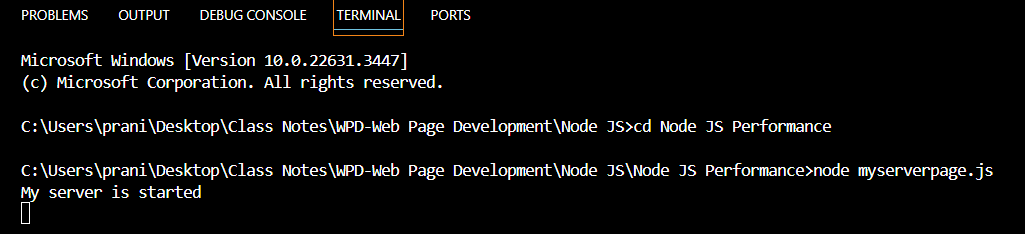




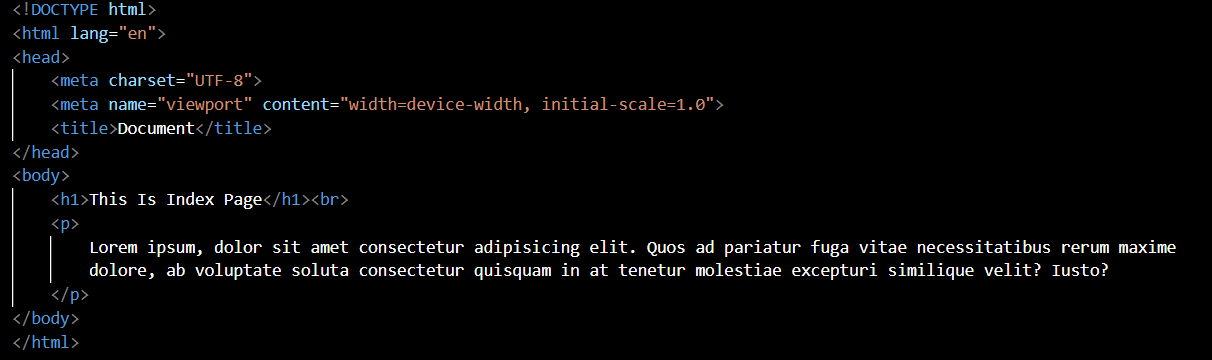


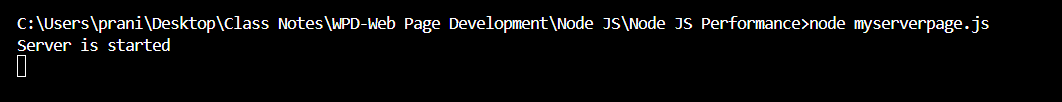
**Basic Code to Create a New Server:**





**How to create HTML Page:**





Run The myserverpage.js code on Chrome by typing: localhost:7770

…………………………………………………………………………………………………………………………

**Express**

**(17/4/2024)**

**DOTENV:**

* It’s a JavaScript package that reads KEY=VALUE from a .env file and sets each pair as env variables.
* It allows us to separate from other in workspace or in open source.
* How to install:

Npm install dotenv -save;

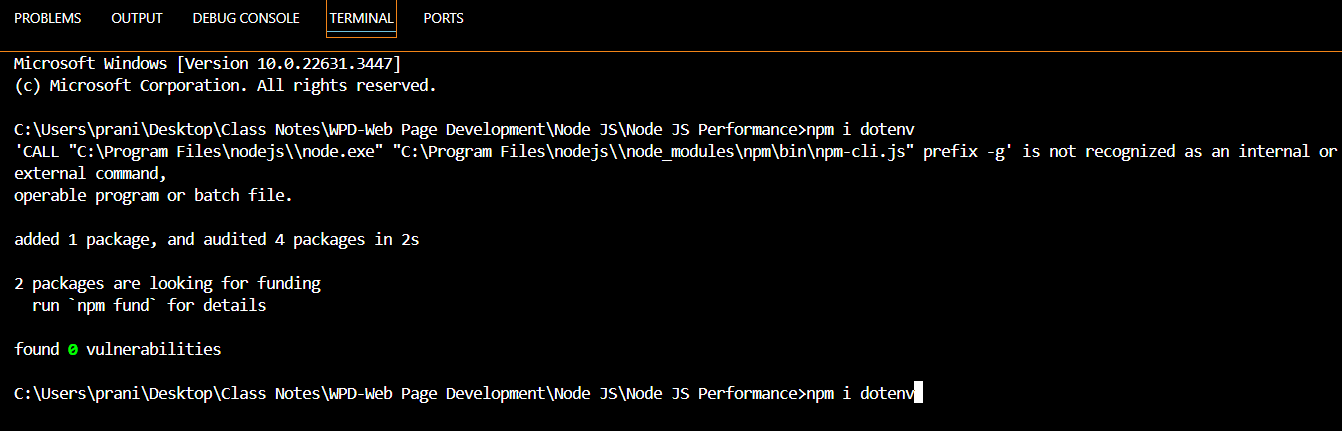
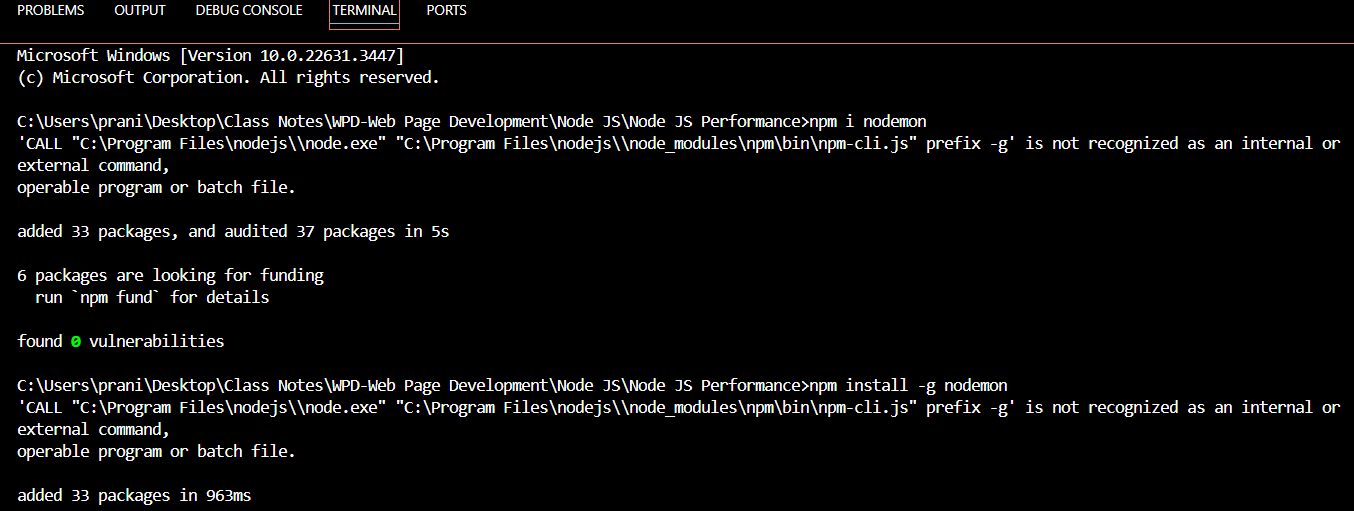
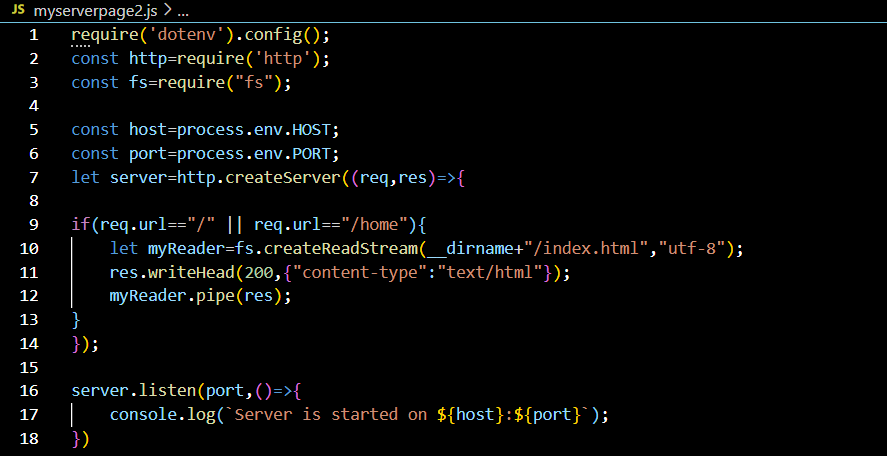
Require(“dotenv”).config ();

**NODEMON:**

Nodemon is a tool that helps to develop node.js based applications, by automatically restarting the node application when file changes in a directory are detected.

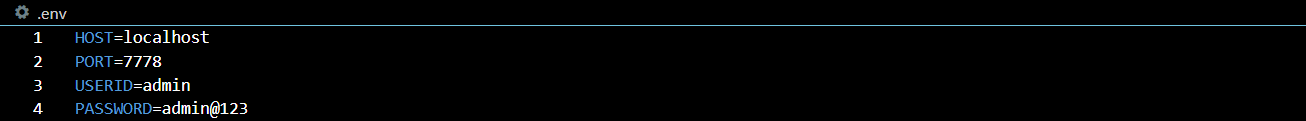
* nodemon does **not** require any additional changes to your code or method of development. nodemon is a replacement wrapper for node. To use nodemon, replace the word node on the command line when executing your script.
* Syntax: npm install -g nodemon

How to Install Nodemon: npm install -g nodemon

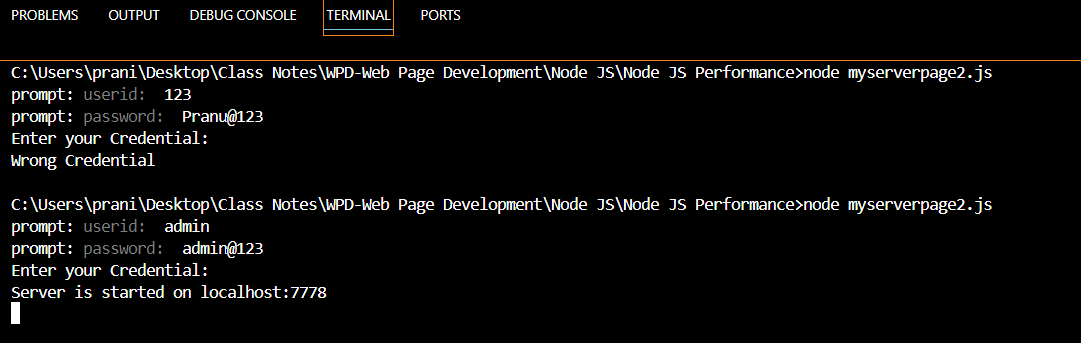


**Prompt:**

* **It allows us to get value terminal.**
* Syntax: npm install --save prompts







**Express JS:**

--Express.js is a web application framework for Node.js, It provides various features that makes web application development fast and easy which otherwise takes more time using only Node.js.

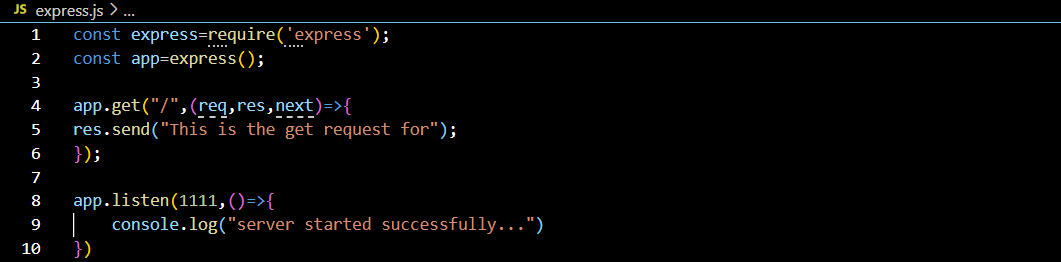
--Express.js provides an easy way to create web servers and render HTML pages for different HTTP requests by configuring routes for our application.

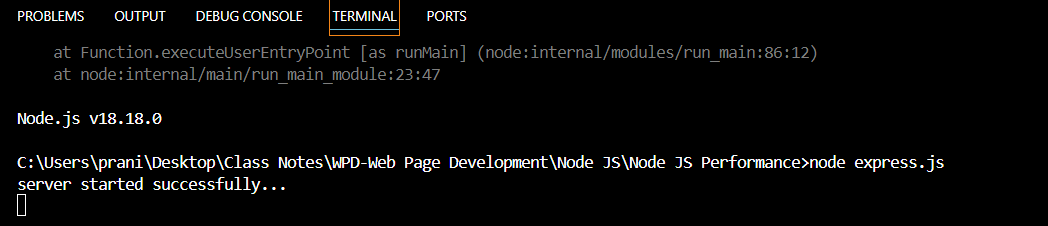
**Advantages:**

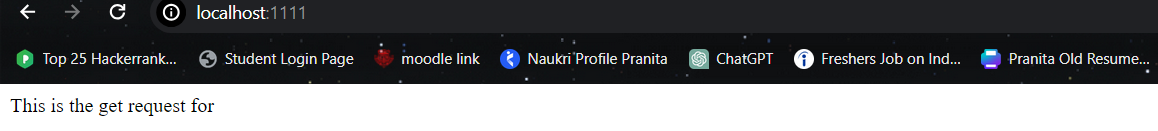
1. Makes Nodes.js web application development fast & easy.
2. Easy to Configure (add external libraries) & Customize (As per our choice we edit)
3. Allows us to define routes of our application based on HTTP method & URL’s.
4. Esay to integrate with different template engines like Jade, EJS etc.

**3Advantages:**

**Installation:** npm I express

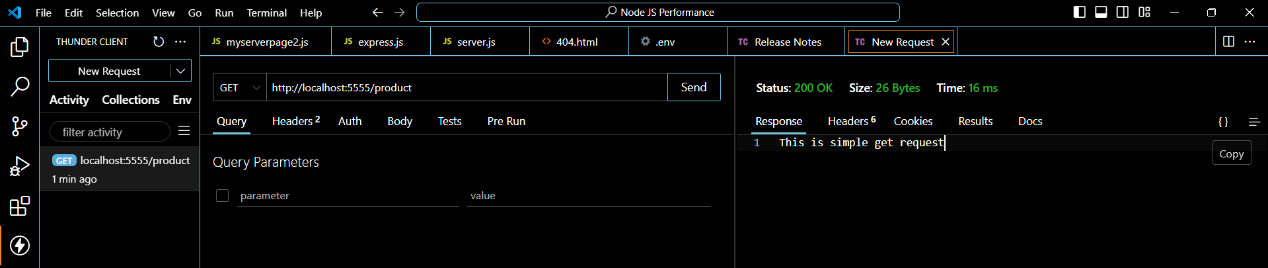
****

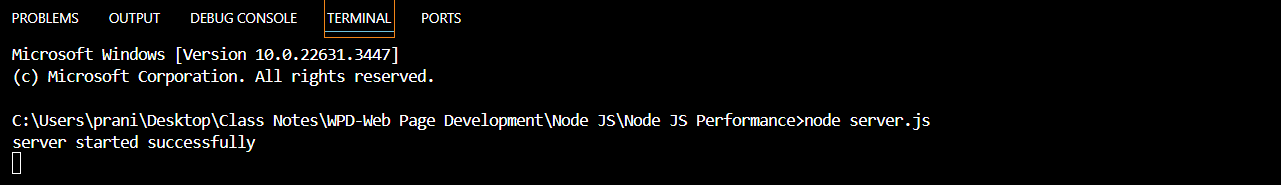
****

****

**How will you set Environment Variable in Node JS: // .**env library

**Create Server.js File and Check it on Thunder client:**

****

****

**Body Parser :**

It allows us to get URL encoded data it is middleware which

* It is easy to serve static files using Built-in middleware in Express.js called “express.static”.
* The Express static middleware is responsible for serving the static assets of an express.js application.
* The express.static() methos specifies that from which to serve all static resources.
* //Setting middleware
* App.use(express.static(\_\_dirname+’public’));

**Web Server Project Steps:**

Step1: package.json create it. // npm init or npm init -y

Step2: Install Express //npm install express --save

Step3: Install dotenv //npm install dotenv --save

Step4: Install nodemon //npm install –save -dev nodemon

**How to Use middleware:**

Express.static()

**How to use Template engine:**

[Express.js](https://expressjs.com/) view engine for [handlebars.js](https://handlebarsjs.com/)

**React.JS**

**(17/04/2024)**

**………………………………………………………………………………………………….**

* App.js is root component in React JS.1
* Components: It is a partial part of react app, which we can reuse multiple times.
* In react app.js is a root component.
* In react we have two types of component:

1. **Function component:** when your main focus is on UI Design in that xase we use it.
2. **Class component:** When we need to write complex logic in that case we use it.

**Note:** When we create component for component name start with capital letter start.

How to display component output individually.:

* To display component output individualy.

To display component output indivisually for first we import component & write component as tag.

**Steps for Installation:**

1. **Open React project in VS:**
2. **Open cmd and check path:** npm start componnentr frim reax

**Open react from react library:**

**ClassComp.js:**

import React,{Component} from “react”

class ClassComp extends Component{

render{ }

{

Return<h2>This is classs componentr</h2>

}

}

Export default ClassComp;

Function.Comp.js:

Function FunctionComp(){

Return<h2>This is function component</h2>

}

Export default functionComp;

**How to display component output individually:**

To display component output individually for first we import component and write component as tag.

**JSX:**

* It is javascript xml.
* It makes react code easy and allows us to write html code in react. In JSX we have to use one parent element to wrap multiple child element. Parent element must be any block element like div, p, h1 to h6 etc.

**PROX:**

* It is parameter or functional parameter which is use to pass data inside component.
* It returns Object type of data.
* To access props in function component for that we need to write : //props.propsname
* To access props in class component for that we need to write : //this.props.propsname

**FunctionComp.js:**

const FunctionComp=(props)=>{

Return<div>

<h2>This is function component</h2>

<p>My name is: <strong>{props.empName}</strong>,

I am:<strong>{props.empPost}</strong></p>

</div>

}

Export default FunctionComp;

**Task:**

Create one function component with name MyDetails containing

myName,myContact, address,gender,education as props with value display its output on browser.

**Task:**

Create one class component with name FriendDetails containing myName, myContact, address, gender, education as props with value display its output on browser.

**Task:** Create one class component with name EmpDetails in that class state having one property with emp containing data for 10 emp with id, name, post, salary, gender, address. Display array of object data into table.

**Task:** MyCounter.js code

**How to use CSS in react:**

1. Internal CSS: We need to write style property with values.
2. External CSS: In this we create a separate css file and that import in component.
3. Module CSS: If you want to write css property for particular component for that we use it.

Its file name must be filename.module.css

We can access it by creating variable of it.

**Babel library:**

Makes & configures your code compatible with new code.

**Conditional rendering:**

Display component output on the basis of condition.

1. If else: In this if condition is true returns if statements otherwise returns else statements.
2. Element as variable: In this we store data in variable and that variable put inside element.
3. Use of ternary operator: It is a short-hand syntax for if else.

**Syntax:** (condition)? true part: false part;

1. Short-circuit: It will execute only when condition is true.

It is denoted by && and basically it is used for Null checking.

**Imp Question:**

**What is Virtual DOM:**

In react instead of manipulating browser DOM directly, react creates virtual DOM in its memory, where it does all necessary changes before making changes in real DOM. That why we get a quick response and react changes only what want to be change. Thats why it is faster.

**How To Render array of object data:**

**Hooks:**

* React hooks allows us to use class component feature in function component.
* By using hooks we can use state in function component also.
* Hook introduces in react version 16.

**Rules to use Hooks:**

1. It only works with function component.
2. It can not be render conditionaly.
3. It always use at top level component.

**E.g:**

1. useState

2. useEffect

3. useContext

4. useRef

5. useReducer

6. useCallback

7. useMemo

8. custome Hook

1. useState: It allows us to use state in function component.

* It allows us two parameters in it: 1. Current state, 2. Function-to-update state
* To use it first we have to import it from react library.
* useState hooks can hold number,string,Boolean,array,object types of data in it.
* Syntax: const[currentState,functionToupdateState]=useState();

**2. useEffect Hook:**

* It is used to perform side effect function in function component.
* It is used to perform operation after rendering of component output.
* Side effects means fetching data from server, Use of Times, DOM manupuation.
* It takes two parameters 1. Callback function 2. Dependency.
* It performs componentDidMount, shouldComponentupdat, ComponentDidUpdate like of class component.
* After component render useEffect hook execute.
* Syntax: useEffect(callbackFunction, dependency)

Case1: When no dependency pass.

* It executes after every render.
* componentDidMount, componentDidupdate

Case2: When we pass dependency as a blank array

* It executes only ones.
* It shouldComponentupdate

Case3: When we pass dependency value.

**19/04/2024: React js**

**HOC (Higher Order Component):**

When we must share common business login with multiple components in that case we use it.

In this we share logic from function component to class component

* Rfc: for function
* Rce for main file

**ErrorBoundary:**

* It is reacting component which is use to handle error in react component tree at view level.
* It contains Two class component lifecycle method:

1. getDerivedStateFromError: It execute automatically when error occur in component.
2. componentDidCatch: It display error log(information).

**Task:** Write HOC Logic for ErrorBoundary component.

**Routing:**

* Navigation of component.
* It displays component or view by writing its routing path name in browser url.
* To perform it in react we must install react-router-dom library.

**Installation link:**

* npm i react-router-dom

**Types of routing:**

1. Default routing: Display component view or output when path is blank.
2. Naming Routing: Display component view by writing its path name in browser URL.
3. Parameterized routing: Write parameter with writing path to display component out.

It will work if user enter wrong path browser url. In this to perform it, We write “\*” in path name.

1. Child Routing: Defining one routing inside another routing.To display child routing output we need to import <outlet /> library in parent routing component. To perform child routinhg, For this we use children property inside parent routing object.
2. Wild card routing: Wild card routing must be last routing in routing formate. // It must be write at last, after this no routing execution happens.

**React.fragment:**

* It is alternative for jsx in react.
* It remove unnecessary element fr
* om DOM.

**Card Having 3 Components:**

1. Header
2. Body
3. Footer

**Material UI:**

* It provide inbuilt component, which is mobile friendly, attractive and responsive.
* It also provide icon.
* **Command for installation of Material UI:**

npm install @mui/material @emotion/react @emotion/styled

* **Icons Installation:**

npm install @mui/icons-material

**Web storage:**

1. **Local Storage:**

Local storage.setItem(“user1”,”admin”) // Set value in local storage

LocalStorage.getItem(“User1”) //get value in Local Storage

localStorage.removeItem(user3) //remove single value from local storage

localStorage.clear() //It remove all values at a time

1. **Session storage:**

sessionStorage.setItem(“user1”,”admin”) // Set value in session storage

sessionStorage.getItem(“User1”) //get value in session Storage

sessionStorage.removeItem(user3) //remove single value from session storage

sessionStorage.clear() //It remove all values at a time

**Axios:**

It allows us to make request from client to server.

**How to Install:**

* Npm I axios
* Json-Server
* Json Database
* How to start Json server and upload db.json file

1. cd src/shared/database
2. json-server - -watch -p 8888 db.json

**Protected Routing:**

**User Login:**

**Use of Concurrently library:** It will execute multiple command simultaneously.

1. How to install concurrently:

**Syntax:** npm i -g concurrently

1. Go to package.json file and put below command in it:

**Syntax:** “json-server”: “cd src/shared/database && json-server –watch -p 8888 db.json”, “start-dev”: “concurrently\” npm start” \ “npm run json-server\””,

1. In terminal write:

Num run start-dev

**Exam Tasks for performance**

Task: 10 Chapter Task =>HTML

Task: Resume Form =>HTML

Task: Train data and Another Table question =>HTML

Task: Position Task =>CSS

Task: Set selected cell background color: =>JS

Task: Write & Implement logic for calculator: //Done =>JS

Task: Design digital clock: //Done =>JS

Task: Display current month and day by using Date class object: //Done =>JS

Task:12-04-2024 //Done =>JS

Task 1 & 2 from 13-04-2024 //Done =>JS

Task: Form Validation after submitting include Password, EmailId, Contact, Gender. //Done =>JS

Task: Form Validation Before submitting include Password. //Done =>JS

Task: Website Using Image,Home,about Service options & Cards =>Bootstrap

Task: Hide,show,Toggle image task =>JQuery

**INTERVIEW QUESTIONS:**

**1.What is schematic element give examples of it.**

Ans: schematic elements can refer to various fundamental constructs or symbols used to represent components, operations, or relationships within code. Here are some examples:

1. **Variables**: Schematic representation of memory locations used to store data. They are typically depicted as identifiers (e.g., **x**, **y**, **count**) in code.
2. **Functions/Methods**: Represented by names followed by parentheses, these encapsulate a sequence of operations to perform a specific task. For example, **calculateArea()** or **printMessage()**.
3. **Control Structures**: Symbols or keywords representing flow control in code, such as loops (**for**, **while**) and conditionals (**if**, **else**, **switch**).
4. **Data Types**: Symbols representing different types of data, such as integers (**int**), floating-point numbers (**float**), strings (**str**), or boolean values (**bool**).
5. **Operators**: Symbols representing mathematical or logical operations, like addition (**+**), subtraction (**-**), multiplication (**\***), division (**/**), equality (**==**), and logical AND (**&&**) or OR (**||**).
6. **Comments**: Not strictly schematic, but they are symbols or annotations used to provide explanations or context within the code. In many programming languages, comments are denoted by specific symbols or keywords (**//** for single-line comments, **/\* \*/** for multi-line comments).
7. **Classes and Objects**: In object-oriented programming, classes and objects are schematic elements representing blueprints and instances of entities, respectively.
8. **Libraries/Modules**: Represented by names or namespaces, these encapsulate reusable code components and functionalities.
9. **Constants**: Represented by identifiers with fixed values, typically denoted using uppercase letters (e.g., **PI**, **MAX\_SIZE**).
10. **Error Handling Constructs**: Symbols representing mechanisms to handle exceptions or errors, such as **try**, **catch**, **throw** in languages with exception handling.

**2.The elements which are having specific meaning.**

Ans: Symantic element: Form, Header, Footer, Nav

Non-Symantic elements: Div,span.

**3.Div and span:**

div (block element) and span (inline elements)

**4.Block and Inline elements:**

Ans:

**Block element:**

-It takes 100% width of its screen

-It always start on new line

-example : h1-h6,p,li,table,form

**Inline element:**

-It takes required width of its

-It never start on new line.

-example: b, i, u, a, img

**5.what are APIs introduced in html5:**

Ans:

* **Geolocation:** it return user Current position.
* **drag and drop :** grabbing an object and drag to another place
* **web worker :** it is small JavaScript code which execute at backend of webpage without affecting its current execution.
* **web storage**:it is use to store user data on browser.

**1.local storage**: it store user data on browser for permanently.

**2.session storage:** it store user data on browser for limited time.

* **SSE(Server Sent Event):** it allows use to get automatic update from server.

**6.new features of html5:**

Ans: HTML5 introduced new semantic elements such as **<header>**, **<footer>**, **<nav>**, **<article>**, **<section>**, **<aside>**, **<main>**, **<audio>** and **<video>**

**7.which are selectors:**

Ans:

**Selectors in CSS:** Used to Selecting an Element.

1. Element Selector: We use Element Name directly in Element selector.

2. Group Selector: We make group of elements and apply common css properties to them.

3. Id Selector (#): When you want to apply CSS Property to single element.

4. Class Selector(.): When we have to apply common bunch of CSS property to an multiple element.

5. Universal Selector: When we have to apply CSS property to all element available in Webpage.

**8.difference between opacity (background transparent karne ke liye 0 to 1) and z index (1 and -1 are values)**

Ans:

**Opacity**:

* **Definition**: Opacity in chart formatting refers to the transparency level of chart elements, such as data points, lines, bars, or background elements.
* **Usage**: Opacity can be applied to chart elements to make them partially transparent, allowing underlying elements or data to show through.
* **Implementation**: Charting libraries typically provide options to adjust the opacity of various chart elements through specific properties or configurations. For example, you might set the opacity of data series, markers, or background elements to achieve a desired visual effect.

**Z-index**:

* **Definition**: Z-index in chart formatting refers to the stacking order or layering of chart elements within the chart canvas.
* **Usage**: Z-index determines which chart elements appear in front of others when they overlap on the chart canvas.
* **Implementation**: Charting libraries often provide options to control the z-index of different chart elements, such as data series, axes, labels, or annotations. This allows developers to specify the layering order of chart elements based on their visual hierarchy.
* **Purpose**: Adjusting z-index allows for precise control over the visual presentation of chart elements, ensuring that important elements are displayed prominently and that overlapping elements are rendered correctly.

**9.what is an array:**

Ans:

It is used to store multiple values with same datatype.

In array values store on Index basis.

Array index always start from 0 index.

To access data outside array for that we write array[index\_number]

Array is Collection of similar type of data.

Array Stores its elements Sequentially, Linearly & Contagiously.

Array index is always start with 0 & Ends with size-1.

Array is denoted by []: Square bracket.

**10.shift and unshift:**

Ans:

The **shift()** method removes the first element from an array and returns that removed element. shift() returns undefined.

E.g:

const array = [1, 2, 3, 4, 5];

const removedElement = array.shift();

console.log(removedElement); // Output: 1

console.log(array); // Output: [2, 3, 4, 5]

The **unshift()** method adds one or more elements to the beginning of an array and returns the new length of the array.

E.g:

const array = [2, 3, 4, 5];

const newLength = array.unshift(1);

console.log(newLength); // Output: 5 (length of the modified array)

console.log(array); // Output: [1, 2, 3, 4, 5]

**11.splice:**

Ans:

**splice()** method in JavaScript is used to change the contents of an array by removing or replacing existing elements and/or adding new elements in place.

Example:

array.splice(start, deleteCount, item1, item2, ...);

**12.for each (It is an array concept)**

Ans:

In JavaScript, **forEach()** is a method available on arrays that allows you to iterate over each element in the array and perform a specified operation on each element. It provides a concise and readable way to loop through arrays without the need for traditional **for** loops.

Syntax:

array.forEach(callback(currentValue, index, array), thisArg);

Example:

const array = [1, 2, 3, 4, 5];

array.forEach(function(currentValue, index, array) {

console.log(`Element at index ${index}: ${currentValue}`);

});

**13.Datatypes in js:**

Ans: There are mainly two datatypes in JS:

1. Primitive datatype: number,string,Boolean,undefine,null;
2. Non-Primitive datatype: array,object;

**14. what is object and what is its purpose:**

Ans: In programming, an object is a fundamental concept used to represent real-world entities, data structures, or abstract concepts. An object encapsulates data and behavior (functions or methods) related to that entity or concept.

In JavaScript, objects are a core component of the language and are used to store collections of key-value pairs, where the keys are strings (or symbols) and the values can be any data type, including other objects, functions, or primitive values.

const person = {

name: "John",

age: 30,

occupation: "Developer",

sayHello: function() {

console.log(`Hello, my name is ${this.name}.`);

}

};

**15.what is map method (for performing manipulation):**

Ans:

In JavaScript, the **map()** method is an array method that allows you to iterate over each element in an array and transform each element based on a callback function, creating a new array with the results of the transformation. The original array remains unchanged. The **map()** method returns a new array containing the results of applying the callback function to each element of the original array.

Syntax:

array.map(callback(currentValue, index, array), thisArg);

**16.filter method (it returns new array who satisfies new condition)**

Ans:

Filter() method creates a new array filled with elements that pass a test provided by a function.

The filter() method does not execute the function for empty elements.

The filter() method does not change the original array.

**17. difference between log in and sign in:**

Ans:

In Login, accessing an existing account by providing credentials (username/password) & In Sign in accessing an existing account by providing credentials (username/password).

Login is specific to accessing an existing user account & Sign in can refer to accessing an existing account or creating a new one.

**18. Difference Between HTTP && HTTPS:**

Ans:

| **Feature** | **HTTP** | **HTTPS** |
| --- | --- | --- |
| Security | Not secure; data transmitted in plaintext | Secure; data encrypted using SSL/TLS |
| Encryption | No encryption | Encryption using SSL/TLS |
| Protocol | Operates over port 80 | Operates over port 443 |
| Authentication | No server authentication | Server authentication using SSL/TLS certificates |
| Trust and Integrity | No guarantee of integrity or trustworthiness | Ensures integrity and trust through encryption |

**19. purpose of super keyword:**

Ans:

In JavaScript, the **super** keyword serves the purpose of allowing a subclass (a derived class) to access and call methods defined on its superclass (parent class). It is primarily used within the context of classes and inheritance.

**20.purpose of c and cpp?**

Ans:

C is more focused on systems programming and low-level tasks, prioritizing efficiency and portability, while C++ extends C with powerful object-oriented programming features, making it suitable for building complex, high-performance software applications.

**21. Various events in JS?**

Ans:

In JavaScript, events are occurrences that happen in the browser or in the DOM (Document Object Model), which can trigger JavaScript functions to be executed.

1. **Mouse Events**:
   * **click**: Triggered when the user clicks on an element.
   * **dblclick**: Triggered when the user double-clicks on an element.
   * **mouseover**: Triggered when the mouse pointer enters an element.
   * **mouseout**: Triggered when the mouse pointer leaves an element.
   * **mousedown**: Triggered when the mouse button is pressed down over an element.
   * **mouseup**: Triggered when the mouse button is released after being pressed down over an element.
   * **mousemove**: Triggered when the mouse pointer moves while over an element.
   * **contextmenu**: Triggered when the right mouse button is clicked to open the context menu.
2. **Keyboard Events**:
   * **keydown**: Triggered when a key is pressed down.
   * **keyup**: Triggered when a key is released after being pressed down.
   * **keypress**: Triggered when a key that produces a character value is pressed down.
3. **Form Events**:
   * **submit**: Triggered when a form is submitted.
   * **change**: Triggered when the value of a form element changes (e.g., **<input>**, **<select>**, **<textarea>**).
   * **input**: Triggered when the value of an input or textarea element changes.
   * **focus**: Triggered when an element receives focus.
   * **blur**: Triggered when an element loses focus.
   * **reset**: Triggered when a form is reset.
4. **Window Events**:
   * **load**: Triggered when the window, document, and all resources have finished loading.
   * **resize**: Triggered when the browser window is resized.
   * **scroll**: Triggered when the user scrolls the page.
5. **Media Events**:
   * **play**: Triggered when media playback begins.
   * **pause**: Triggered when media playback is paused.
   * **ended**: Triggered when media playback reaches the end.
   * **timeupdate**: Triggered when the playback position changes.
6. **Drag and Drop Events**:
   * **dragstart**: Triggered when the user starts dragging an element.
   * **drag**: Triggered when an element is being dragged.
   * **dragend**: Triggered when the user stops dragging an element.
   * **dragenter**: Triggered when a dragged element enters a drop target.
   * **dragover**: Triggered when a dragged element is over a drop target.
   * **dragleave**: Triggered when a dragged element leaves a drop target.
   * **drop**: Triggered when a dragged element is dropped on a drop target.

**22. Server side languages? //JAVA, PHP, .Net, Python:**

Ans:

Server-side languages are programming languages that are used to develop the backend logic of web applications. Unlike client-side languages such as JavaScript, which run in the user's web browser, server-side languages execute on the web server. T

1. **PHP**: PHP (Hypertext Preprocessor) is one of the most widely used server-side scripting languages. It is especially popular for web development and is often embedded within HTML to create dynamic web pages. PHP is known for its simplicity, flexibility, and wide range of built-in functions.
2. **Python**: Python is a versatile programming language that can be used for both server-side and client-side development. It is known for its readability and ease of use. Python frameworks like Django and Flask are commonly used for building web applications.
3. **Ruby**: Ruby is another popular server-side language, often used with the Ruby on Rails framework. Ruby on Rails emphasizes convention over configuration and follows the Model-View-Controller (MVC) architectural pattern. It is known for its productivity and elegant syntax.
4. **JavaScript (Node.js)**: While JavaScript is primarily known as a client-side language, it can also be used for server-side development using the Node.js runtime environment. Node.js allows developers to build scalable and high-performance web applications using JavaScript on the server side.
5. **Java**: Java is a widely-used programming language known for its platform independence and robustness. Java-based frameworks like Spring and Hibernate are commonly used for building enterprise-level web applications.
6. **C# (ASP.NET)**: C# is a language developed by Microsoft and is often used for building web applications using the ASP.NET framework. ASP.NET provides a powerful environment for building scalable and secure web applications on the Microsoft platform.
7. **Go (Golang)**: Go is a statically typed, compiled language developed by Google. It is known for its simplicity, performance, and concurrency support. Go is gaining popularity for building web servers and microservices.

**23. What is Client & Server Side Application. //from Node.JS**

Ans:

**Client-Side Application**:

* **Definition**: A client-side application runs on the user's device (such as a web browser) and is responsible for rendering the user interface and interacting with the user.
* **Technology**: Client-side applications are typically built using HTML, CSS, and JavaScript.

**Server-Side Application**:

* **Definition**: A server-side application runs on the server and is responsible for processing requests from clients, executing business logic, and generating dynamic content.
* **Technology**: Server-side applications can be built using various server-side programming languages and frameworks.

24. CRUD Operations:

25. Login in react

26. Vitual DOM, HOC, Props

**Mon:** JQuery

**Tue:** AJAX

**Wed:** NodeJS

**Thu:** React

**Fri:** React

**Sat:** Exam

**HR Questions:**

1. Tell Me About Yourself:

* Full Name
* Where are you from
* Education: Skill set
* Family background
* Hobby
* Special activities

**Good morning, sir. I am Pranita Bhosale & I am from Sangli. I did my diploma in Shivaji Polytechnic college sangola with score 95% marks. I just completed BTech from Walchand College of engineering Sangli in CSE Branch. With first class distinction.**

**Talking about my technical skills I like C, CPP, Web Development, Basics of JAVA, Data structure. In my extra-curricular activities I have done some web development projects like IOT Innovation center portal for my college then Smart agroculture system & One android based project i.e. Sangeet Riyaz.**

**I have great learning ability to learn new things. I am hosnest, Punctuate & Hard worker. Currently My short term goal is to achieve a good job in well company & my long term goal is to build my carrier and support the organization also.**

1. What Is Your Salary Expectation:

**Exam Important:**

* Grid
* form validation
* replace
* string
* before and after submitting
* html, CSS, JS, bootstrap, AJAX, JQuery