Module-32

API :

(Application programming language)

- দিয়ে সার্ভার থেকে ডাটা লোড করে আমার ওয়েবসাইট এ দেখাবে |

JSON: javascript object notation.

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax.

1.কোনো একটা জাভা স্ক্রিপ্ট অবজেক্ট থাকলে তা আমি স্ট্রিং করতে পারি |একবার স্ট্রিং হয়ে গেলে পর্ প্রপার্টি এক্সেস করতে পারবো না|

```
const shop = {
   name: 'bappa',
   adddress: 'ball ',
   profit: 1300,
   product: ['cricket ', 'pepsi', 'pen'],
   isCostly: false
};
const shopStringified = JSON.stringify(shop)
```

2. এখন স্ট্রিং থেকে এক্সেস করতে চাইলে,
JSON.parse() করতে হবে |

```
// normal js e convert kore felci
const converted = JSON.parse(shopStringified);
console.log(converted);
```

Fetch জিনিষটা বুঝার চেষ্টা করলাম. Where we can load data.

```
/ first fetch the url
// second set the fetch url in json()
// finally conslog the json.
```

```
fetch('https://jsonplaceholder.typicode.com/todos/1')
    .then(response => response.json())
    .then(json => console.log(json))
```

Output :
{userId: 1, id: 1, title: "delectus aut autem", completed: false}

dynamicallt কোন ওয়েবসাইট থেকে নিজের **add** করতে চাইলে, ডাটা লোড করতে পারি সেই সাথে **website** দেখতে পারি।

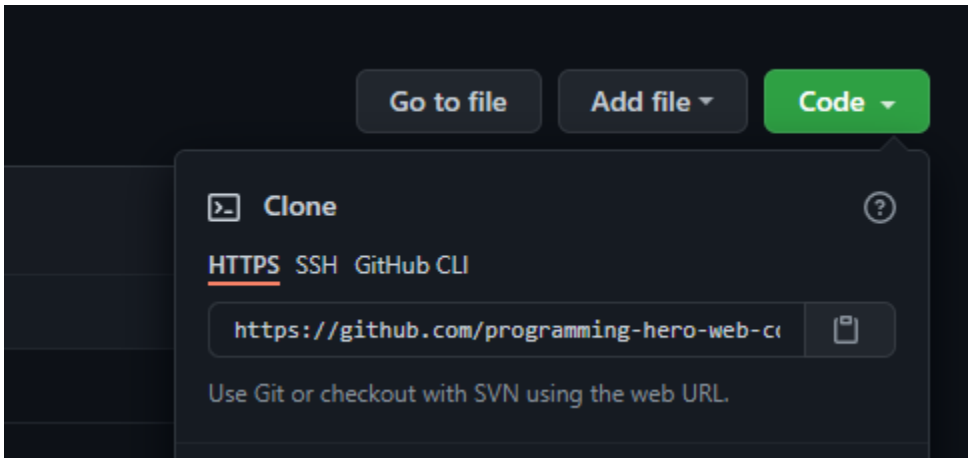
```
function loadUsers() {
    fetch('https://jsonplaceholder.typicode.com/users')
        .then(response => response.json())
        .then(dataName => displayUserNames(dataName))
}

function displayUserNames(dataName) {
    const ul = document.getElementById('users')
    for (const user of dataName) {
        const list = document.createElement('li');
        list.innerText = `username is : ${user.name}`;
        ul.appendChild(list);
    }
}
```

Milestone-6

Git clone and update repository

#1st. HTTPS:



https://github.com/programming-hero-web-course2/honda-cbr-bootstrap-assignment-bappasahabapi.git

#2nd:PS

F:\pro-hero-projects\assignments\assignment-6> git clone --bare https://github.com/programming-hero-web-course2/honda-cbr-bootstrap-assignment-bappasahabapi.git

#3rd

git push --mirror

https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git

All code is here

PS F:\pro-hero-projects\assignments\assignment-6> git clone --bare https://github.com/programming-hero-web-course2/honda-cbr-bootstrap-assignment-bappasahabapi.git
Cloning into bare repository 'honda-cbr-bootstrap-assignment-bappasahabapi.git'... remote: Enumerating objects: 53, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (41/41), done. eceiving objects: 13% (7/53)

remote: Total 53 (delta 16), reused 45 (delta 11), pack-reused 0 52% (28/53), 14.01 MiB | 1.58 MiB/s Receiving objects: 100% (53/53), 14.80 MiB | 1.54 MiB/s, done.
Resolving deltas: 100% (16/16), done.
PS F:\pro-hero-projects\assignments\assignment-6>
PS F:\pro-hero-projects\assignments\assignment-6> cd .\honda-cbr-bootstrap-assignment-bappasahabapi.git\ PS F:\pro-hero-projects\assignments\assignment-6\honda-cbr-bootstrap-assignment-bappasahabapi.git> git push --mirror https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git
Enumerating objects: 53, done.

```
Counting objects: 100% (53/53), done.
Delta compression using up to 4 threads
Compressing objects: 100% (36/36), done.
Writing objects: 100% (53/53), 14.80 MiB |
831.00 KiB/s, done.
Total 53 (delta 16), reused 53 (delta 16),
pack-reused 0
remote: Resolving deltas: 100% (16/16), done.
To
https://github.com/bappasahabapi/phero-hond
a-cbr-bootstrap.git
* [new branch]      main -> main
PS
F:\pro-hero-projects\assignments\assignment-
6\honda-cbr-bootstrap-assignment-bappasahab
api.git> git remote -v        origin
https://github.com/programming-hero-web-co
urse2/honda-cbr-bootstrap-assignment-bappas
ahabapi.git (fetch)        origin
```

https://github.com/programming-hero-web-course2/honda-cbr-bootstrap-assignment-bappasahabapi.git (push)

PS F:\pro-hero-projects\assignments\assignment-6\honda-cbr-bootstrap-assignment-bappasahabapi.git> git remote set-url origin https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git

PS F:\pro-hero-projects\assignments\assignment-6\honda-cbr-bootstrap-assignment-bappasahabapi.git> git remote -v

origin https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git (fetch)

origin https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git (push)

PS F:\pro-hero-projects\assignments\assignment-

6\honda-cbr-bootstrap-assignment-bappasahab api.git>

API : (32/33/34/35)

Module-32

(Application programming language)

- দিয়ে সার্ভার থেকে ডাটা লোড করে আমার ওয়েবসাইট এ দেখাবে |

JSON: javascript object notation.

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax.

1.কোনো একটা জাভা স্ক্রিপ্ট অবজেক্ট থাকলে তা আমি স্ট্রিং করতে পারি |একবার স্ট্রিং হয়ে গেলে পর্ প্রপার্টি এক্সেস করতে পারবো না|

```
const shop = {
   name: 'bappa',
   adddress: 'ball ',
   profit: 1300,
   product: ['cricket ', 'pepsi', 'pen'],
   isCostly: false
};
const shopStringified = JSON.stringify(shop)
```

2. এখন স্ট্রিং থেকে এক্সেস করতে চাইলে,
JSON.parse() করতে হবে |

```
// normal js e convert kore felci
const converted = JSON.parse(shopStringified);
console.log(converted);
```

 # Fetch জিনিষটা বুঝার চেষ্টা করলাম. Where we can load data.

```
/ first fetch the url
// second set the fetch url in json()
// finally conslog the json.
```

```
fetch('https://jsonplaceholder.typicode.com/todos/1')
    .then(response => response.json())
    .then(json => console.log(json))
```

Output :
{userId: 1, id: 1, title: "delectus aut autem", completed: false}

 # dynamicallt কোন ওয়েবসাইট থেকে নিজের **add** করতে চাইলে,
ডাটা লোড করতে পারি সেই সাথে **website** দেখতে পারি |

```
function loadUsers() {
   fetch('https://jsonplaceholder.typicode.com/users')
      .then(response => response.json())
      .then(dataName => displayUserNames(dataName))
}

function displayUserNames(dataName) {
```

```
    const ul = document.getElementById('users')
    for (const user of dataName) {
        const list = document.createElement('li');
        list.innerText = `username is : ${user.name}`;
        ul.appendChild(list);
    }
}
```

Module -- 33

#visit : https://jsonplaceholder.typicode.com/

Visit : https://api.kanye.rest/

```
const loadQuotes = () => {
    fetch('https://api.kanye.rest/')
        .then(response => response.json())
        .then(data => displatQuote(data))
}
// loadQuotes();
const displatQuote = (lines) => {
    const quoteElement = document.getElementById('quote');
    quoteElement.innerText = lines.quote;
}
```

Visit : https://randomuser.me/

```
const loadBuddies = () => {
    fetch('https://randomuser.me/api/?results=5')
        .then(res => res.json())
```

```
        .then(data => displayBuddies(data))
}
loadBuddies()

const displayBuddies = (data) => {
    console.log(data.results[0].name.first);
}
```

Get here access using array because the type of the api is an array.

Ex-2:

```
const loadBuddies = () => {
    fetch('https://randomuser.me/api/?results=5')
        .then(res => res.json())
        .then(data => displayBuddies(data))
}
loadBuddies();

const displayBuddies = (data) => {
    // console.log(data.results[0].name.first);
    const friends = data.results;

    const buddiesDiv = document.getElementById('buddies');
    for (const buddy of friends) {
        // console.log(buddy.email);

        const p = document.createElement('p');
        p.innerText = buddy.email;
        buddiesDiv.appendChild(p);
```

```
    }
}
```

Ex:3: make the inner text dynamic using backtag ``

```
// make inner text dynamic
        const p = document.createElement('p');
        p.innerText = `Name: ${buddy.name.first}   ${buddy.name.last}
                    Email: ${buddy.email} `;
        buddiesDiv.appendChild(p);
```

Visit: https://restcountries.eu/

Here the data is given in array of array.

If its arrey the we also can use **mapping().**

Mapping use no return

For of use / forEach loop use korte pari

1st way:

```
const loadCountries = () => {
    fetch('https://restcountries.eu/rest/v2/all')
        .then(response => response.json())
        .then(data => displayCountries(data)
        )
}
loadCountries();
```

```
const displayCountries = (countries) => {
    for (const country of countries) {
        console.log(country);
    }
}
```

2nd way is more appropriate:

```
const loadCountries = () => {
    fetch('https://restcountries.eu/rest/v2/all')
        .then(response => response.json())
        .then(data => displayCountries(data)
        )
}
loadCountries();
const displayCountries = (countries) => {

    countries.forEach(country => {
        console.log(country);
    })
}
```

Add multiple element in a div in js api Example

```
const loadCountries = () => {
    fetch('https://restcountries.eu/rest/v2/all')
        .then(response => response.json())
        .then(data => displayCountries(data)
        )

}
loadCountries();
const displayCountries = (countries) => {
    const countriesDiv = document.getElementById('countries');
    countries.forEach(country => {
        console.log(country);
        const div = document.createElement('div');
        div.classList.add('country')
// add css class country using js

        const h1 = document.createElement('h1');
        h1.innerText = country.name;
        div.appendChild(h1);
        const p = document.createElement('p');
        p.innerText = country.capital;
        div.appendChild(p);

        countriesDiv.appendChild(div);
    });
}

```

#Most easiest way is bagtag

```
const loadCountries = () => {
    fetch('https://restcountries.eu/rest/v2/all')
        .then(response => response.json())
        .then(data => displayCountries(data)
        )
}
loadCountries();
const displayCountries = (countries) => {
    const countriesDiv = document.getElementById('countries');
    countries.forEach(country => {
        console.log(country);
        const div = document.createElement('div');
        div.classList.add('country') // add css class country using js
        div.innerHTML = `
        <h3>Country Name:${country.name}</h3>
        <h4> Captal is: ${country.capital}</h4>
        <p> Area is:  ${country.area}</p>
        `;
        countriesDiv.appendChild(div);
    });
}
```

Project food search

Meal db api

First :

1. 1st get the input field text .
2. The call the fetch part
3. Then the write display function() to show them in my website.

Second:

```
//04. creat and append div
```

- Make div dynamically by inner html
  And inject this div to our searchResult part of the website.
- Create dynamic url based on click and display data. That means show the single data.
- **Remove previous data results.**
  searchResults.textContent= ‘ ‘;
  searchResults.inner.HTML= ‘ ‘;

Module 34

5. Condition check

Double equall (==) → match the value
Triple equall (==) → match both value and type.

6. Variables

Local scope

Block scope (let , const)

Global scope

Hoisting (var)

- ☐ Global leaking

7.

What is **Closure** in js**?(Interview question)**

একটা ফাংশন থেকে যদি আমি আরেকটা ফাংশন রিটান করি তাহলে ,তার নিজের একটা ক্লোজার তৈরী করে, বা প্রাইভেট রেফারেন্স তৈরী করে that is called Closure. Or
Encapsulation.

8.

call back function()

কোন একটা ফাংশনের প্যারামিটার হিসেবে আরেকটা ফাংশনকে পাঠানো হলে, এবং ভিতরে থেকে যদি কল করা হত , সেইটাই call back function()

Country details

```
// 1. user amr site e visit korbe
// 2. user kichu search dibe
// 3. amra search result dekhabo
// 4.buttton e click korle detais dibo
```

Milestone-7

Module -38

search on enter button click. → google it.(43--5--4)

array.length = 0 , means the array is now empty [ ].

For each / for of loop are used in finding element of an array .

For in loop is used in finding element of an object
.

BOM --- browser object model
#alert: just works as modal. Just give one option “ ok”
#confirm: it gives two option ,”ok” ,“ cencel”
That means it gives question?
#prompt: question + box is given.
Response ta console e print kore dekhabe.

#
https://developer.mozilla.org/en-US/docs/Web/API/Location
Console e :
location llikbo.
window.location
location.reload() → Refresh kora button
location.assign() → oi location e niya jabe.
Location.href
location.hash

location.search
&t=20s → er por start hbe.

Local storage vs Session Storage valo vabe dekhbo.

module-40.5

Find the documentation of console tab utilities api.
What is accessibility?
what is a call stack?
What is the purpose of regular expression?
#What is the purpose of the lighthouse of Chrome dev tab.
Go to the source tab and add a conditional break point.

Issue Tracker bug fix.

Module - 42

#

Template string = `back tag er vitore dynamically kichu likha. `

arrow function ⇒

spread operator(...)

#very very important

map = kichu return korbe.

forEach = kichu returen korbe na .

filter = sortto sappekhe kaj kore. Ja ja match korbe tar sob gulai dita dibe.

find = jeita match korbe tar prothom ta just dita dibe.

Module 43

Debugging part ta dekte hbe.

Module - 51

React Bootstrap

React bootstrap ta valovabe dekhbo.

#Conditional formatting: ei jinish ta valovabe dekhbo.

Material Ui

Install process :

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

// with yarn

yarn add @mui/material @emotion/react @emotion/styled
```

React Routing 53

Amazon project

Routing most commonly used in App.js

React notes M-44

All codes are tested in practise-app folder

#**Component** : similar the pattern , different the data.

#**Routing** : pathe location is called the routing.

ex:https://www.amazon.com/wedding?ref_=registries_subnav

#**State** : means the changing status .

Working folder : src is the only working folder.

React

Terminal e type:(to create react app)

1.npx create-react-app bapi-app
2.cd bapi-app
3. npm start

In react we have to work import **two method**

1. import React from '**react**';
2. import ReactDom from '**react-dom**';

 # *React* run the html code in **render function**.
render → means display

ReactDom.render(); → takes three parameters.
ReactDom.render(কি দেখাবো , কোথায় দেখাবো , callback function);

#1→ ***Example : single line***

```
ReactDom.render(
<h1> hello bappa</h1> ,
document.getElementById('root')
);
```

---------xxxxxx---------

<h1> hello bappa</h1> → কি দেখাবো

document.getElementById('root') → কোথায় দেখাবো

"<> </>" → this is called Rect Frangement.

→ ***Write multiple line***

```
ReactDom.render(
<>
<h1> hello bappa</h1>
<h1> hello bappa</h1>
</> ,
document.getElementById('root')
);
```

Module - 45

Visual studio code react emmet ? google search dibo.

JSX stands : **for JavaScript XML**. It is simply a syntax extension of JavaScript

#Bsisc six things :

1. JSX,
2. Component: similar in look , different in data .
   - আমাদের নিজেদের নামের ক্লাস নেইমগুলো catital letter দিয়ে শুরু হবে ।
     example:(Human)

   - Compone k dynamic korte amra "props "--> means property use kori.
     Mane argument ba patameter use korte pari .

#Props amra pathabo **html** er moto ko
re and receive it like **javaScript** er.

```
const [variable ,function ] = useState(0);
```

```
useEffect(function, parameter)
```

Module --1

Save the file **.html** extension

#add favicon:

```
<title> Module-1</title>
  <link rel="icon" href="/Module-1/image/big-sur.jpg" type="image/x-icon"/>
```

#কোন লিখাকে বোল্ড করতে

```
<b>This is bold text</b
<strong>This is strong tag </strong>
```

#add image:

```
 <img src="" alt="">
```

#small

```
<small>small tag la la la la  </small>
```

#anchor tag

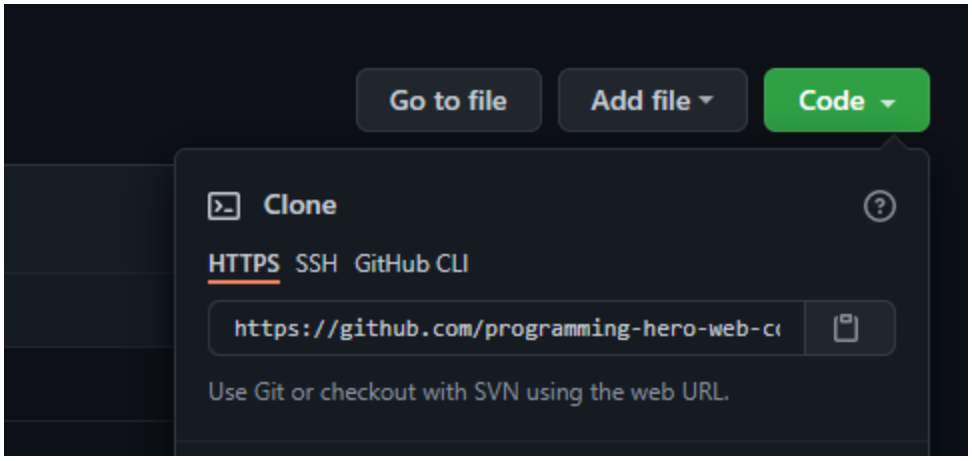
```
<a
href="https://www.google.com/?sa=X&ved=0ahUKEwjsqabPnMXrAhWTdn0KHUPpAT0QOwgC">
go to google
</a>
```

#Form input:

```
<form action="""">
    <input type="text"> This is text input </input><br><br>
    <input type="text"> <br> <br>
    <input type="password" name="" id="">
    <input type="checkbox" name="" id="">
    <input type="radio" name="" id="">
    <input type="file" name="" id="">
    <input type="color" name="" id="">
     <br>
</form>
```

Git clone and update repository

#1st. HTTPS:



https://github.com/programming-hero-web-course2/honda-cbr-bootstrap-assignment-bappasahabapi.git

#2nd:
PS F:\pro-hero-projects\assignments\assignment-6> git clone --bare https://github.com/programming-hero-web-course2/honda-cbr-bootstrap-assignment-bappasahabapi.git

#3rd
git push --mirror

https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git

All code is here

```
PS F:\pro-hero-projects\assignments\assignment-6> git clone --bare https://github.com/programming-hero-web-course2/honda-cbr-bootstrap-assignment-bappasahabapi.git
Cloning into bare repository 'honda-cbr-bootstrap-assignment-bappasahabapi.git'...                                    remote:
Enumerating objects: 53, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (41/41), done. eceiving objects:  13% (7/53)
remote: Total 53 (delta 16), reused 45 (delta 11), pack-reused 0                    52% (28/53), 14.01 MiB | 1.58 MiB/s Receiving objects:
```

```
100% (53/53), 14.80 MiB | 1.54 MiB/s, done.
Resolving deltas: 100% (16/16), done.
PS F:\pro-hero-projects\assignments\assignment-6>
PS F:\pro-hero-projects\assignments\assignment-6> cd .\honda-cbr-bootstrap-assignment-bappasahabapi.git\                PS F:\pro-hero-projects\assignments\assignment-6\honda-cbr-bootstrap-assignment-bappasahabapi.git> git push --mirror https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git
Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Delta compression using up to 4 threads
Compressing objects: 100% (36/36), done.
```

```
Writing objects: 100% (53/53), 14.80 MiB |
831.00 KiB/s, done.
Total 53 (delta 16), reused 53 (delta 16),
pack-reused 0
remote: Resolving deltas: 100% (16/16), done.
To
https://github.com/bappasahabapi/phero-hond
a-cbr-bootstrap.git
* [new branch]      main -> main
PS
F:\pro-hero-projects\assignments\assignment-
6\honda-cbr-bootstrap-assignment-bappasahab
api.git> git remote -v        origin
https://github.com/programming-hero-web-co
urse2/honda-cbr-bootstrap-assignment-bappas
ahabapi.git (fetch)        origin
https://github.com/programming-hero-web-co
urse2/honda-cbr-bootstrap-assignment-bappas
ahabapi.git (push)        PS
```

F:\pro-hero-projects\assignments\assignment-6\honda-cbr-bootstrap-assignment-bappasahabapi.git> git remote set-url origin https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git
PS F:\pro-hero-projects\assignments\assignment-6\honda-cbr-bootstrap-assignment-bappasahabapi.git> git remote -v origin https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git (fetch)
origin https://github.com/bappasahabapi/phero-honda-cbr-bootstrap.git (push)
PS F:\pro-hero-projects\assignments\assignment-6\honda-cbr-bootstrap-assignment-bappasahabapi.git>

প্রোগ্রামিংয়ের ছয়রত্ন কি তুমি কি জানো:

১. কিভাবে let, const দিয়ে ভেরিয়েবল লিখতে হয়। কখন কোনটা ইউজ করে তুমি কি জানো

২.১ কিভাবে কন্ডিশন লিখতে হয়। ছয় রকমের কন্ডিশন (>, <, ===, !==, >=. <=) কোনটা কোন জিনিসের জন্য ইউজ করবে। এছাড়াও && বা ।। দিয়ে কিভাবে একাধিক কন্ডিশন এর মধ্যে দুইটাই ফুলফিল করতে হবে আবার দুইটার যেকোন একটা ফুলফিল করতে হবে সেটা কিভাবে করবে

২.২. এক বা একাধিক কন্ডিশন দিয়ে কিভাবে if-else লিখে। আবার কখন if-else-if - else লিখে। সেই রকম একটা উদারহণ চিন্তা করে তুমি লিখে ফেলো

৩. array কিভাবে ডিক্লেয়ার করে। array এর মধ্যে length, index, push, pop, indexOf, includes এইগুলা কিভাবে কাজ করে। কোনটা দিয়ে কি করে? সেগুলা কি তুমি জানো?

৪. দুইটা বেসিক বেসিক লুপ এর মধ্যে for loop তোমাকে জানতেই হবে। while লুপটাও দেখে রাখতে পারো। যদিও আমরা এই দুইটা লুপ ই কম ইউজ করবো। তাও কখনো লাগলে যেন তুমি বুঝে ফেলতে পারো।

৫. function একটা অবশ্য জিনিস। বিশেষ করে সিম্পল একটা ফাংশন কখন ডিক্লেয়ার করতে হয়। কখন ফাংশন থেকে return করে। আর কিভাবে ফাংশন এর মধ্যে parameter নিতে হয়।

৬. আখেরি রত্ন হচ্ছে Object তাই কোন একটা অবজেক্ট কিভাবে ডিক্লেয়ার করে। সেখান property কিভাবে কিভাবে একসেস করা যায়। এছাড়াও অবজেক্ট এর প্রপার্টি এর ভ্যালু হিসেবে কিভাবে array, object ইউজ করা যায়।

ES ৬ রিলেটেড সাতটা জিনিস তোমাকে জানতে হবে

১. একটা টেম্পলেট স্ট্রিং দিয়ে একটা স্ট্রিং ভেরিয়েবল ডিক্লেয়ার করো। সেটার মধ্যে অবজেক্ট এর প্রপার্টি এর মান কিভাবে বসায় সেটা জানতে হবে। বিশেষ করে নেস্টেড অবজেক্ট আছে সেটার প্রপার্টি থেকে। বা কোন একটা অবজেক্ট এর প্রপার্টি array সেই array থেকে ভ্যালু এনে কিভাবে টেমপ্লেট স্ট্রিং এর মধ্যে বসাতে পারবে ।

২. স্প্রেড অপারেটর (...) কিভাবে কাজ করে। বিশেষ করে একটা array কে কপি করে নতুন করে array বানাবে এবং সেখানে একটা উপাদান যোগ করবে। আবার একটা উপাদান কে বাদ দিয়ে বাকি সব উপাদানকে কিভাবে যোগ করবে (filter ইউজ করে)

৩.১. শূন্য প্যারামিটার ওয়ালা একটা অ্যারো ফাংশন লিখবে যেটা ৯ রিটার্ন করবে।

৩.২. এক প্যারামিটার ওয়ালা একটা অ্যারো ফাংশন ডিক্লেয়ার করবে। এই ফাংশনের কাজ হবে যে প্যারামিটার নিবে সেটাকে ১২ দিয়ে গুণ করে গুণফল রিটার্ন করবে।

৩.৩ দুই, প্যারামিটার ওয়ালা একটা অ্যারো ফাংশন লিখবে। এই ফাংশনের কাজ হবে যে দুইটা প্যারামিটার ইনপুট নিবে। সেই দুইটা প্যারামিটারকে যোগ করে যোগফলকে চার দিয়ে ভাগ করে ভাগফল রিটার্ণ করে দাও।

৩.৪ একাধিক লাইন ওয়ালা অ্যারো ফাংশন লিখো। সেটাতে দুইটা প্যারামিটার নিবে। ওই arrow ফাংশনটা হবে অনেকগুলা লাইনের। সেখানে প্রত্যেকটা ইনপুট প্যারামিটার এর সাথে ৫ যোগ করবে তারপর যোগফল দুইটাকে আবার গুণ করবে। ক্যামনে করবে সেটা চিন্তা করে বের করার চেষ্টা করো।

৪. সিম্পল একটা জাভাস্ক্রিপ্ট অবজেক্ট এর কোন একটা প্রোপার্টিকে ভেরিয়েবল হিসেবে ডিক্লেয়ার করার জন্য destructuring ইউজ করো। array এর destructuring করবে আর সেটা করার জন্য তুমি এরে এর সেকেন্ড পজিশন এর উপাদান কে destructuring করে 'balance' নামক একটা ভেরিয়েবল এ রাখবে।

৫. shorthand দিয়ে অবজেক্ট কিভাবে ডিক্লেয়ার করে। {a , b } স্টাইলে।

৬. ফাংশন এর মধ্যে ডিফল্ট প্যারামিটার কিভাবে ডিক্লেয়ার করে

৭. অপশনাল চেইনিং কি জিনিস। সেটা কখন কিভাবে ইউজ করে। না জানলে গুগলে সার্চ দাও

ব্রাউজার API সম্পর্কে চারটা জিনিস

১. লোকাল স্টোরেজ, সেশন স্টোরেজ কোনটা কখন ইউজ করবে। কিভাবে ইউজ করবে

২. location API কিভাবে ইউজ করবে ব্রাউজারে

৩. history API কিভাবে ইউজ করে

৪. একদম প্রাথমিক স্টেপ হিসেবে jsonplaceholder এর ওয়েবসাইট থেকে ডাটা fetch করে সেটাকে কনসোল এ দেখাতে হবে। ধরো তুমি তাদের ওয়েবসাইট এ photos এর API এর লিংক কোনটা সেটা জাভাস্ক্রিপ্ট দিয়ে কোড করে সেই ডাটা কনসোল এ দেখতে পারতেছো কিনা। তারপর কয়েকটা কার্ড বানাবে (হতে পারে বুটস্ট্রাপ এর কার্ড) সেগুলা আবার তুমি html দিয়ে ওয়েবসাইট এ ছবি এবং ছবির নিচে ছবির টাইটেল দেখাবে।

আরো পাঁচটা জিনিস জানতে হবে।

১.১ অনেকগুলা সংখ্যার একটা array হবে। তারপর তোমার কাজ হবে array এর উপরে map ইউজ করে। প্রত্যেকটা ২ দিয়ে গুণ করে গুণফল আরেকটা array হিসেবে রাখবে। পুরা কাজটা এক লাইনে হবে।

১.২. জাভাস্ক্রিপ্ট এ array এর map, forEach, filter, find কোনটা দিয়ে কি হয়। সেটার একটা সামারি লিখে ফেলো।

২. ternary অপারেটর কি। এইটা দিয়ে শর্টকার্টে কিভাবে if-else লিখে

৩. লজিক্যাল এন্ড (&&) আর লজিক্যাল or (।।) এই দুইটা সম্পর্কে হালকা ধারণা

৪. JSON এর stringify এবং parse কখন কোনটা ইউজ করে

৫. ++, --, +, +", +=, -= এইগুলা কি জিনিস। কোনটা দিয়া কি করে সেটা বুঝলেই হবে। তাছাড়া active = !active এইটা এর মানে কি।

৬. Object.keys, Object.values জিনিসগুলা জানা থাকলেও ভালো।

Cover Letter

Bappa Saha
Thakurgaon,Rangpur, Bangladesh
Contact No: +8801303246040
Email: bappasaha161@gmail.com
03 December, 2021

Dear Hiring Manager,

Thank you for the opportunity to apply for the MERN stack web developer position. Reviewing the job description, it's clear that you're looking for a candidate that is familiar with React, Bootstrap, Node.js, NoSQL with the responsibilities associated with this role, and can perform them confidently. Given these requirements, I am certain that I have the skills to successfully do the job adeptly and perform above up to the mark.

I've 1 year of experience in MERN stack web development and system design and I'm self-motivated, energetic, enthusiastic and ever-ready to learn new technology. I love being in this line of work because I'm passionate about technology, design & innovation. I'm confident to ensure the highest level of professionalism and commitment to my job.

Reviewing my resume, I hope you will agree that I'm the type of competent and competitive candidate you are looking for. I look forward to elaborating on how my specific skills and abilities will benefit your organization.

Thank you for your consideration, and I look forward to sharing more about my skills and experiences. Feel free to contact me at +8801303246040 or bappasaha161@gmail.com

Sincerely
Bappa Saha

All interview questions

HTML

01.What are the differences between html4 and html5?

> html5 has many new tag which was not present in html4 like audio, video canvas tag etc. Better error handling is introduced in html5.

02. What are semantic tags in html? Give me some examples.

A semantic element clearly describes its meaning to both the browser and the developer. Example: header , nav ,section, footer article etc. for example nav has its meaning that indicate navigation section.

Examples of non-semantic elements: and - Tells nothing about its content. Examples of semantic elements.

03.What is the purpose of Article, div, section, nav, aside?

<article>
The article element represents a section of content that forms an independent part of a document or site; for example, a magazine or newspaper article, or a blog entry.
<section>
Used for "sections" of a web page that have grouped, related content.
<aside>
Similar to <section>, but for side bars. Hence the name "aside".
*** ***
Used for sections of a page that don't have a more appropriate tag.
<nav>
represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation section is menubar.

04 Why will you use Meta tag?
metadata is used by browsers how to display content or reload page, search engines (keywords), and other web services.
used to specify character set, page description, keywords, author of the document, and viewport settings.

05.What is the difference between inline, inline-block, and block?
Inline: here if we set height width properties that does not work. It takes space what its content need.
Inline-block: i can set both height and width with values.
block: it takes the whole width and starts with a new line.

06.Difference between strong, b, bold, em, i?
Strong: represent the importance
b: indicating the style of bold
i:indicating the style of italic format.
em: will force the word like emphasis

07.What are properties and attributes in HTML?

attributes provide additional information about elements. Example : href, src , img,url etc.

Properties: what we write inside a oppninig tag is property like name, value,placeholder etc.

08.What is a Viewport?

The visible portion of the screen .It varies with the device - it will be smaller on a mobile phone than on a computer screen.

09.what are void elements in HTML?

that do not have closing tags or do not need to be closed are void elements.
For Example

09. What is the significance of <head> and <body> tag in HTML?

<head> tag provides the information about the document.
<body> tag defines the body of the HTML document.

10. Is it possible to change an inline element into a block level element?

Yes, it is possible using the “display” property with its value as block to change the inline element into a block-level element.

11.Difference between link tag <link> and anchor tag <a>?

The anchor tag <a> is used to create a hyperlink to another webpage or to a certain part of the webpage and these links are clickable.
Link ag <link> defines a link between a document and an external resource and these are not clickable.

12. How to include javascript code in HTML?

Using <script > tag

13.When to use scripts in the head and when to use scripts in the body?

When we use event-triggered functions or jquery library then we should use them in the head section.
When we write content on the page it should be placed is side the body at the bottom part.

14. Difference between SVG and Canvas HTML5 element?

SVG works better with a larger surface.
Canvas works better with a smaller surface.

15. What are the significant goals of the HTML5 specification?

Introduction of new element tags to better structure the web page such as <header> tag.

Top 20 Css Interview Questions

01.What Flex layout? Difference Flex and grid layout?

Ans: Flexbox is mainly meant for 1-dimensional layouts while Grid is meant for 2-dimensional layouts.
This means Flexbox can work on either row or columns at a time, but Grids can work on both. Flexbox, gives you more flexibility while working on either element (row or column).

02.Explain CSS position property? What are some differences between absolute position and relative position?

Ans:The CSS position property defines the position of an element in a document. This property works with the left, right, top, bottom and z-index properties to determine the final position of an element on a page.
position: relative places an element relative to its current position without changing the layout around it, whereas
position: absolute places an element relative to its parent's position and changing the layout around it.

03.What is a box model? And what are the different elements of a box model?
Ans:
Ans: The box model is used to determine the height and width of the rectangular box. The CSS Box consists of Width and height (or in the absence of that, default values and the content inside), padding, borders, margin.

04.What is a Hover effect? What is the purpose of the active class ?
Ans:
Hover effects are a part of most site designs. They can be simple — like enlarging an image — or, they can trigger multi-step animations. ... Hover effects bring interactivity and motion to a design, making for a more dynamic web experience.
The active class is applied to the navigation element the user is currently viewing. In the case of your given code, the user is viewing the the profile. It will serve as a guide or reminder to where in the website the visitor is in

5. What are the different types of Selectors in CSS?

Ans:Universal selector : *{ }
Id selector : #id { }
Class selector: .class{ }
Child Combinator:

6. What is CSS Specificity?

: CSS specificity is a score or rank that decides which style declaration has to be used to an element. (*) this universal selector has low specificity while ID selectors have high specificity. There are four categories in CSS which authorize the specificity level of the selector.

- Inline style
- IDs
- Classes, Attributes, and pseudo-classes.
- Elements and pseudo-elements.

7. What is a CSS Preprocessor? What are some benefits of Sass?

Ans:A css preprocessor is a tool used to extend the basic functionality of default vanilla CSS through its own scripting language.
It helps us to use complex logical syntax like – variables, functions, mixins, code nesting, and inheritance to name a few, supercharging your vanilla css.

Sass advantages:

- It's css syntax friendly.
- It offers variables for whatever you want
- Sass facilitates you to write clean, easy and less CSS in a programming construct.
- It contains fewer codes so you can write CSS quicker.
- It is more stable, powerful, and elegant because it is an extension of CSS. ...
- It is compatible with all versions of CSS.

8. What is a Pseudo element? What is pseudo-class?
Ans:
It is a feature of CSS which is used to style the given parts of an element.

```
selector::pseudo-element {
property:value;
}
```

It is a class that is used to define a special state of an HTML element.
This class can be used by styling an element when a user snooped over it and also it can style an HTML element when it gets the focus.

```
selector:pseudo-class {
property:value;
}
```

9. How will you use media queries to make a website responsive?

Ans:It uses the @media rule to include a block of CSS properties only if a certain condition is true.
If the browser window is 600px or smaller, the background color will be lightblue:

```
@media only screen and (max-width: 600px) {
  body {
        background-color: lightblue;
  }
}
```

10. How will you make font size responsive?

Ans:Use em or rem. One of the best practices to make the responsive text on the web is to use relative units like rem and em . From the type scale that we have defined, we can see the size in em on the right side of its px unit. The em is a unit which equals to currently specified font-size.

11. Explain the three main ways to target elements?

Ans: A, using the tag , you can target body ,p ,span
B,by using class
C. by using id

12. Difference between px, em rem?

Ans:

1em represents 16px .
1rem represents 16px .

Px : it is not scalable . it is fixed value

Em: unit is relative to the font size of the element being styled. This is also affected by inherited values from the parent elements unless it is explicitly overridden by a px unit which is not subject to inheritance.

Rem: unit is relative to the font size of the root (html) element.

13. Explain the concept of Tweening?

Ans:It is mainly used for creating animation.
weening is the process in which we create intermediate frames between two images to get the appearance of the first image which develops into the second image.

14. Why gradients are used in css?

Answer: It is a property of CSS which allows you to display a smooth transformation between two or more than two specified colors.
There are two types of gradients that are present in CSS. They are:

- Linear Gradient
- Radial Gradient

15. Difference between margin and padding?

margin is the property by which we can create space around elements. We can even create space to the exterior defined borders.
padding is the property by which we can generate space around an element's content as well as inside any known border.

16. How does CSS actually work (under the hood of browser)?

- The browser converts *HTML* and *CSS* into the *DOM (Document Object Model)*. The DOM represents the document in the computer's memory. It combines the document's content with its style.
- The browser displays the contents of the DOM.

17. How does Z index function??

Overlapping may occur while using CSS for positioning HTML elements. Z index helps in specifying the overlapping element. It is a number which can be positive or negative, the default value being zero.

18. Differentiate between sass vs scss?

- SASS is based on indentation and SCSS(Sassy CSS) is not.
- SASS uses .sass extension while SCSS uses .scss extension.
- SASS doesn't use curly brackets or semicolons. SCSS uses it, just like the CSS.

Sass syntax:

```
$font-color: #fff
$bg-color: #00f

#box
        color: $font-color
        background: $bg-color
```

Scss syntex:

```
$font-color: #fff;
$bg-color: #00f;
```

```
#box{
        color: $font-color;
        background: $bg-color;
}
```

19.what is VH/VW (viewport height/ viewport width) in CSS?

It's a CSS unit used to measure the height and width in percentage with respect to the viewport. It is used mainly in responsive design techniques. The measure VH is equal to 1/100 of the height of the viewport. If the height of the browser is 1000px, 1vh is equal to 10px. Similarly, if the width is 1000px, then 1 vw is equal to 10px

20.What does the :root pseudo-class refer to?

The :root selector allows you to target the highest-level "parent" element in the DOM, or document tree. It is defined in the CSS Selectors Level 3 specification.

JavaScript interview questions

1. How does JavaScript work?
or What is the JavaScript Event Loop?
Or Though JavaScript is single-threaded, how does it handle concurrent work?
Or Is JavaScript Single-threaded or multi-threaded?
Or Is JavaScript Synchronous or asynchronous?

Ans: javaScript is asynchronous single threaded . that means it has only one call stack
That means handle one task at a time .

2. How does JavaScript code is executed in Browser?
Ans:
Whenever a JavaScript program is run an execution context is created. In this phase, javascript allocates the memory to all the variables and functions present in the program.

3. What are the differences between "==" and "===" ?
Ans: The == operator checks only the value , where === checks both value and its type.

8**. What is a callback function?**
Ans: A function passed into another function as an argument . that means it is called by another function.
5. When will you return something from a function?
Or How will you return more than one value from a function?
Ans: when the value is needed form another function to get a calculation.

6. Tell me about bind, call and apply.

Or How many arguments does call apply bind take?
Ans: **call()** method calls a function with having this value and arguments provided individually.
The **apply()** method calls a function with having this value, and arguments provided as an array.
bind() method returns a new function, where the value of “this” keyword will be bound to the owner object, which is provided as a parameter.

7. What is a Closure in JavaScript? How does it work?
Ans:closures are created when a variable needs to define outside from the current scope and it is accessed from within some inner scope.
provide a better, creative, and expressive writing code for the developers
Like : length()--> returns the length of a strings.

8. What does the “this” keyword indicate in JavaScript?
Ans:
this keyword is needed for defining object. It points to the particular object. We can apply
These rules to the this keyword in order to know which object refers to this keyword.
Global scope, object scope

9. What is Event bubbling in js? Or How does event delegate work in JS?
Ans: dom element nesting refers to the bubbling. Simply means if the handle of a child is clicked it also works for the parent elements means parents handler also clicked too.

10. Explain hoisting in JavaScript.

Ans: default behavior of code execution. Used at the top of the scope . simply variable functions declared at the top part .

11. What is a recursive function function
Ans: A function which is called by itself

12. Difference between undefined and null
Ans:
Undefined means the variable is declared but the value is not still assigned.
Null simply means no value . In array it means no object.

13. What are the different data types in JavaScript?
Or Primitive data type and non-primitive data type
Ans: number , string, boolean, object , array.
primitive types are number and boolean data types. Non-primitive types like strings.

14. What is DOM
Ans:The document object model is a programming api for html and xml documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

15. Is JavaScript a static type or a dynamic type?
Or How will you know the type of a JavaScript variable?
Ans: dynamic type

16. What is the use of NaN function?
Ans: NaN means not a number . if the argument or parameter is not a number the the function return NaN.

17. What is undeclared variables ?
Ans: undeclared means the variable is not declared yet or the variable is not exist in the program.

18. What is the purpose of array slice in java script?
Ans: the slice() method returns the selected elements of an array.

19. How can you convert the string of any base to integer/ float in JavaScript?
Ans: using the parseInt() for integer and using parseFloat() for float value.

20. What is anonymous functions in javascript?
Ans: An anonymous function is a function that does not have a valid name.

21.How to detect the operating system on the client machine?
Ans: using the navigator app version
22.What is the use of Void (0)?
Ans: used for preventing page refreshing.

23. What is the use of the blur function?
Ans: used to remove the focus from the specific like (input field) object.

24. What is memoization?
Ans: optimization technique of storing previously executed computations.

25. What is the use of a constructor function in javascript?
Ans: Constructor functions are used to create objects in javascript.

25 ES6 interview question

1. **What ES6 features did you use?**

- Some ES6 features I used,

  Let and const keywords, Default Parameters, Arrow functions, Object Literals, Rest and spread operators, Destructuring assignment, Promises, and many more.

2. **What are the differences between var, let, and const?**

   we use **var** to declare variables. It can be re-assign after assign a variable.

   let: is changeable.

   const: the const keyword is not changeable .

3. **Why will you use default parameters?**

   - To reduce the error of undeclared parameters we will use default parameters in a function.

4. **How does the Spread operator work?**

   - The spread operator is commonly used to make shallow copies of JS objects or arrays. The spread operator … is used to expand or spread an iterable or an array.

   Denoted as "..."

5. **Difference between class and object.**

   - Class is used as a template for declaring and creating the objects, And an object is an instance of a class.
   - When a class is created, no memory is allocated but Objects are allocated memory space whenever they are created.

6. **What is a Prototype chain?**

   - All objects in JavaScript are instances of Object. That means all the objects in JavaScript inherit the properties and methods from Object.prototype. This is called Prototype chaining.

7. **Explain Call by value vs call by reference**

    - **Call by value:**Let declare a variable named “**a**” and store an integer value in “**a**”. a=10;Now the variable **“a”** stores 10 and has an address location where that primitive value sits in memory.

      Now, suppose we copy the value of **“a”** in **“b”** by assignment (**a=b**). Now, **“b”** points to a new location in memory, containing the same data as variable **“a”**.

      Thus, a=b=5 but both points to separate locations in memory.

      This approach is called **call by value** where 2 variables become the same by copying the value but in 2 separate spots in the memory.

      EX: let a = 10;

      ```
      let b;

      b = a;

      a = 3;

      console.log(a) // 3;

      console.log(b) // 10;
      ```

    - **Call by reference:** Let’s say, we have an object stored in the variable “**a**”. The variable stores the location or the address where the object lives. Now we set **b=a**. Now that new variable “**b**” instead of pointing to a new location in the memory, points to the same location where “**a**” does. No new object is created, no copy is created. Both the variables point to the same object. This is like having 2 names.

      This is call by reference. It behaves quite differently from by value. All objects interact by reference.

8. **What is the scope of JavaScript?**

    - Scope in JavaScript refers to the current context of code, which determines the accessibility of variables to JavaScript.

9. **What is a Higher-order Function?**

- A higher-order is a function that either takes a function as an argument or returns a function.

10. **What is API? Difference between Get vs post?**

    - API **(An application programming interface)** is a connection between computers or between computer programs**.**
    - **Get:** This API is used to get or fetch any data from the data collection
    - **POST:** This API is used to post any data to a server which is sent from the client-side.

11. **Difference between local storage and Session storage.**

    - The storage capacity of session storage is lower than the local storage.
    - Client can only read local storage. They can't read session storage.

12. **What are cookies? And why will you use it?**

    - Cookies are small text files with small pieces of data — like a username and password. Cookies are used to identify a user's computer as the user uses a computer network.

13. **What is object-oriented programming?**

    - Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields, and code, in the form of procedures.

14. **Difference between Array vs LinkedList.**

    - The major difference between Array and Linked list regards to their structure. Arrays are index based data structure where each element associated with an index. On the other hand, Linked list relies on references where each node consists of the data and the references to the previous and next element.
    - Array is a collection of elements of similar data type. And Linked List is an ordered collection of elements of same type, which are connected to each other using pointers.

15. **How will you debug a JavaScript application?**

- Using chrome dev-tools I can debug a JavaScript application.

16. **Define ECMAScript.**

- It is the specification that is defined in the ECMA-262 standard to create a general-purpose scripting language

17. **What is the arrow function, and how to create it?**

- Arrow functions are the shorthand notation to write ES6 function. The definition of the arrow function consists of parameters, followed by an arrow (=>) and the body of the function.

EX: const function (x,y)=>{ body of the function }

18. **Discuss the for...in loop.**

- It is similar to for loop that iterates through the properties of an object. It is useful when we require to visit the properties or keys of the object.

19. **Define Babel.**

- Babel is one of the popular transpilers of JavaScript. It is mainly used for converting the ES6 plus code into the backward-compatible version of JavaScript that can be run by previous JavaScript engines.

20. **What is set?**

- Set is the collection of new values. In Set, there shouldn't be any duplicate value. All the values should be unique. These values can be primitive types or object references.

21. **What is Destructuring in ES6?**

- In ES6 destructuring was introduced to extract data from arrays and objects into an individual variable. It allows you to extract the smaller fragment from objects and arrays.

22. **Define Map in ES6.**

Map is a new way to represent the data in key-value pairs. Map is ordered, and it remembers the insertion order of the keys.

23. Define undefined?

- In Javascript, undefined means, changable is affirmed but doesn't hold value.

24. What are the new String methods introduced in ES6?

There are four string methods introduced in ES6 that are listed as follows:

- string.startsWith()
- string.endsWith()
- string.includes()
- string.repeat()

25. List the new Array methods introduced in ES6?

There are many array method available in ES6, which are listed below:

- Array.of()
- Array.from()
- Array.prototype.copyWithin()
- Array.prototype.find()
- Array.prototype.findIndex()
- Array.prototype.entries()
- Array.prototype.keys()

- Array.prototype.values()

Top 25 React Interview Questions

Question-1: What is reactjs? Tell us about the advantages and disadvantages of using react js.
Answer– React is a JavaScript library for building user interfaces.

Advantages:

1. With the help of Virtual Dom it increases the performance of application's.
2. JSX made code easy to read and write.
3. It renders both server side and client side.

Disadvantages:

1. React Js shows only the library but not the full framework.
2. The code complexity increases with inline templating and JSX.

Question-2: What is JSX? How does it work?
Answer– JSX is a shorthand for JavaScript XML(Extensible Markup Language). It is a kind of file used by React to understand HTML easily. It is a file which utilizes the expressiveness of JavaScript along with HTML like template syntax.

Question-3: What is Virtual dom? What are the differences between virtual and real dom?
Answer– Virtual Dom is an in-memory representation of real DOM. Virtual dom kept the memory of Real Dom which showed in UI. When we change anything in our code to update anything in the UI, it stores in Virtual Dom. Then Virtual Dom calculates the difference between Real Dom and Virtual Dom. When the calculation is done the Virtual Dom updated Real Dom with only the things that have actually changed.

Question-4: Differences between props and state?

Answer– Props are components which contain a single value or set of objects which passed components to another component. Props passed data downward from parent component to child component. And State is a component that holds some information that may change. They are used for internal communication inside a component.

Question-5: What is the purpose of useState? When and why will you use it?
Answer– When I need to store any value which will change, then I will use useState.
Example– const [name, setName] = useState();
Question-6: What is prop drilling?
Answer– Props drilling is a way of data transfer which passes data to the downward. When we need to transfer data from one component to another component we can send by props drilling. Passing data from the parents component to children component is simply called props drilling.

Question-7: Difference between useEffect and useState?
Answer– When we need to fetch data from outside of the component we use sideEffect or useEffect, and need to store that data which we collect from outside we use useState. Which data we get from using useEffect that can be changed, those data we store in useState.

Question-8: What other hooks have you used other than useState and useEffect.
Answer– useParams, useContext, useRef

Question-9: Tell us about React Component lifecycle.
Answer– Basically React Component lifecycle means what happens to react application when it is used. The series of events that happen from the mounting of a React Component to its unmounting. Lifecycle of React

- ➔ **Mount**: Birth of a component
- ➔ **Update**: Growth of a component
- ➔ **Unmount**: Dead of a component

Question-10: What is the purpose of a custom hook? How will you create a custom hook? Give us an example.

Answer– Custom Hooks offer the flexibility of sharing logic that wasn't possible in React components before. You can write custom Hooks that cover a wide range of use cases like form handling, animation, declarative subscriptions, timers, and probably many more we haven't considered.

There are 5 steps–

- Create an app “hooks-demo” using below command:
  npx create-react-app hooks-demo
- Make your app is working:
- Create a file “useOrderCount.js” to write our custom hook logic:
- Use “useOrderCountHook” in our app:
- Run the app and see the browser using localhost:3000:

Question-11: What is the most challenging task you have accomplished using react?

Answer– Most challenging task I’ve accomplished using react is function drilling by props. Because most of the time, still now I can’t understand properly how a function works in any component whereas the function is declared in other components.

Question-12: What are Redux and its uses?

Answer– Redux is an open-source JavaScript library that is used to manage the application state. Some redux components are Store, Action and Reducer.

Question-13: Do you know about React native?

Answer– Yes I know a little bit about React native. React native is a cross platform framework which is used to develop mobile software/application. It can be used for android as well as ios. It makes code easy to understand, and probably 95% mobile applications use React native.

Question-14: What is a higher-order component? Give us an example.
Answer– A higher order component is a kind of function which takes a component as a parameter and returns it as a new component. A higher order component function accepts another function as an argument. The **map** function is the best example to understand this.

Question-15: How would you optimize a react js application?
Answer– In here, we will discuss five important ways to optimize the performance of a React application, including pre-optimization techniques. These include:

- Keeping component state local where necessary
- Memoizing React components to prevent unnecessary rerenders
- Code-splitting in React using dynamic import()
- Windowing or list virtualization in React
- Lazy loading images in React

Question-16: What is React Fiber?

Answer– React Fiber is the reimplementation of the core algorithm in React v16. Maing goal of React Fiber is to increase suitability of areas like animation, layout etc.

Question-17: What is Lifting State Up in React?

Answer– When several components need to share the same changing data, then we lifted up the shared data to the nearest parents. From the parents we can see that changing data to the children as props. As a result at a same time all component will able to get the same data. This process is called Lifting State Up.

Question-18: What is context?

Answer– Context is a process topasses data through a component tree without having to pass props down manually at every level of the tree.

Question-19: Why do we use JSX?

Answer–

- It is faster than regular javaScript.
- It makes it easier to make templates.
- Most of the errors can be found at compilation time, and so on.

Question-20: What is Props?

Answer– Props is the short form of "Properties". It helps us to pass data from the parent to the child components throughout the application.

Question-21: What is an event in React?

Answer– An event is an action which triggers as a result of the user action or system generated event like a mouse click, pressing a key etc.

Question-22: What is create-react-app?

Answer– Create React App is a tool introduced by Facebook to build React applications.

Question-23: How to pass data between react components?

Answer– With the help of props we can send data from parents to children.

Question-24: What are child props in React?

Answer– In React Children/childs are props that help to send components to another component as data, just like any other props.

Question-25: What are portals in React?

Answer– Portals is a recommended way to render children into a DOM node that exists outside the DOM hierarchy of the parent component.

Top 6 Nodejs & MongoDB Interview Questions

Question– 1: What is Nodejs?
Difference between Nodejs and javaScript.
Or, is Node js blocking or non-blocking?

Answer– Node.js is a server-side platform built on google chrome's JavaScript Engine (V8 Engine).
It is an open source server environment and the most positive thing about Node.js, it is free for developers.
It runs various platforms on Windows, Unix, Linux, Mac OS X etc. It uses JavaScript on the server.

Question– 2: Why did you use Node and Mongo with your React project?
Answer–
I use NodeJS with my React project for those reasons– React and Node both are JavaScript basis platforms.

I can execute them both server-side and client-side.
I use Node because I get to inject V8 engine performance into my React app. NodeJS is a real-time and data-intensive React app.
And I also used NodeJS with React because I'm comfortable and easily learn with both of them.

Question– 3: What are the differences between sql and nosql databases?

Answer– Some key difference I've found in my developing experience are listed below–

- ★ SQL databases are relational, NoSQL databases are non-relational.
- ★ SQL databases use structured query language and have a predefined schema. Whereas, NoSQL databases have dynamic schemas for unstructured data.
- ★ SQL databases are vertically scalable, NoSQL is horizontally scalable.
- ★ SQL databases are table-based, while NoSQL databases are document based.
- ★ SQL databases are better for multi-row transactions, while NoSQL is better for unstructured data like documents or JSON.

Question– 4: What have you done with mongodb?

Answer– I used mongoDB in my React app to store my structured or unstructured data.
Because in real-time applications users may provide data and this data should be stored in private secure places.
I chose this private secure place as MongoDB.
we show the content in the UI, which data is needed to store anywhere instead of using in the react app code for better performance.

Question– 5: Have you worked on website hosting?

Answer– I worked on website hosting on a free basis like github.com, netlify.com and firebase. It is very useful and helpful for the junior web developer for practicing and learning purposes. It also has many limitations. **Shared Branding–** Along with

Ads, some free web hosting companies will treat websites as a way to promote their brand. **Lack of customer support** is another limitation.

Question– 6: What is the syntax to create a collection and to drop a collection in MongoDB?
Answer–

- ➔ Syntax to create collection in MongoDB is db.createCollection (name, options)
- ➔ Syntax to drop collection in MongoDB is db.collection.drop()

Video LInk I have watched: https://youtu.be/4Cc75EC1llE

React Native Questions

Question No 1: How Different is React-native from ReactJS?
Answer: React Js is a javaScript library that supports both the front-end and back-end. It is a popular JavaScript library built by Facebook. It is used to develop mobile as well as web-based applications. On the other hand, React Native is a cross-platform mobile framework that uses the ReactJS framework. React Native is used to make mobile

applications for android & ios. Using React Native a developer can build mobile applications for both android & ios writing the same code.

Question No 2: What are the advantages of native apps over hybrid apps?
Answer: Native apps are basically created targeting specific devices like android & ios. That means native apps need to create separately for android & ios. Otherside Hybrid apps are independent. One app is suitable for both android & ios devices. Developer code for one time which is enough for both devices.
Advantages of Native Apps Over Hybrid Apps:

- The performance of native is high and fast than hybrid apps.
- Native apps guarantee the best user experience and performance.
- In the case of security, Native apps provide the best security in apps than hybrid apps.

Question No 3: How will you call an API in React Native?
Answer: By using the fetch method [fetch()] we can call an API in React Native. Once an API is called to the server, the server gets back with a response. Our React Native app needs to handle this response from the server.

Question No 4: Name core Components in React Native and the analogy of those components when compared with the web.
Answer:

React Native UI Component	Web Analog
<view>	
<Text>	<p>
<Image>	<img>
<ScrollView>	
<TextInput>	<input type = “text”>

Question No 5: What is the InteractionManager and how is it used? Why is it important?
Answer: In React Native which helps to run JavaScript animation smoothly is known as InteractionManager. It allows long-running work to be scheduled after any interactions/animations have completed.

InteractionManager is very important for React Native to run our animation smoothly.

React Native has two threads. We use this InteractionManager so that we can ensure that our function is executed successfully after that animation occurs, so that we do not drop any frame in UI threads.

5 Company Linkedin Link Where I Applied

1. **https://www.linkedin.com/company/cdon-com_2/**
2. **https://www.linkedin.com/company/northvolt/**
3. **https://www.linkedin.com/company/collabhero/about/**
4.
5. **https://www.linkedin.com/company/websolutionfirm/**