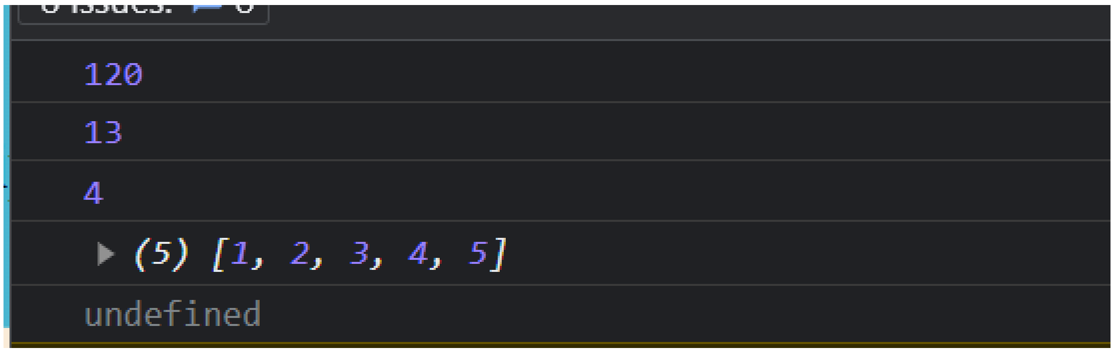
717821E202_Ajay S

Week 3:

1. Recursion and stack
1. Recursion and stack: o Task 1: Implement a function to calculate the factorial of a number using recursion. o Task 2: Write a recursive function to find the nth Fibonacci number. o Task 3: Create a function to determine the total number of ways one can climb a staircase with 1, 2, or 3 steps at a time using recursion. o Task 4: Write a recursive function to flatten a nested array structure. o Task 5: Implement the recursive Tower of Hanoi solution.

```
let sum=1;
function findFactorial(num)
{
    sum*=num;
    const perform=num-1;
    if(perform>0)
    {
        findFactorial(perform);
    }
}
findFactorial(5);
console.log(sum);

function findFibo(num)
{
    if(num<=1)
    {
        return num;
    }
    return findFibo(num-2)+findFibo(num-1);
}
console.log(findFibo(7));

function totalWays(way)
{
    if(way==0)
    {
        return 1;
    }
    if(way<0)
    {
        return 0;
    }
    return totalWays(way-1)+totalWays(way-2)+totalWays(way-3);
}
```

```
console.log(totalWays(3));

function flatten(arr) {
    return arr.reduce((acc, cur) => acc.concat(Array.isArray(cur) ?
flatten(cur) : cur), []);
};

const arr = [[1,2],[3,[4,[5]]]];
const flattened = flatten(arr);
console.log(flattened);

function towerOfHanoi(n, from_rod,  to_rod,  aux_rod)
{
        if (n == 0)
        {
            return;
        }
        towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
        towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);
    }
    var N = 3;
    console.log(towerOfHanoi(N, 'A', 'C', 'B'));
```



```
120
13
4
▶ (5) [1, 2, 3, 4, 5]
undefined
```

2. JSON and variable length arguments/spread syntax:

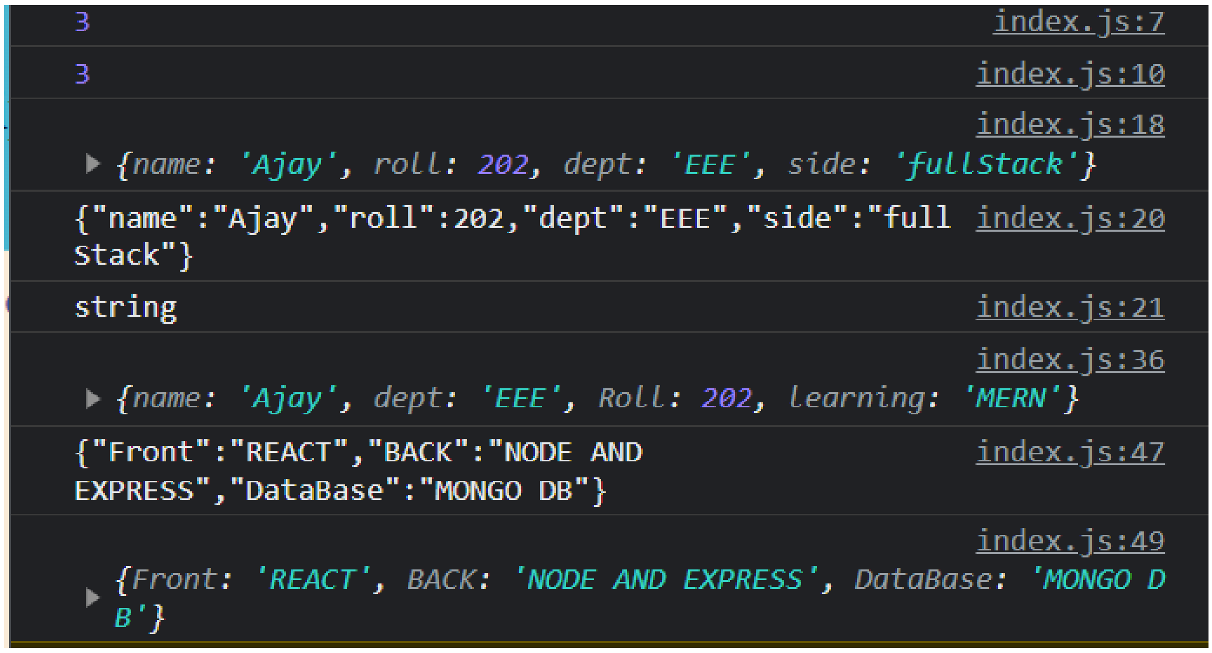
Task 1: Write a function that takes an arbitrary number of arguments and returns their sum. Task 2: Modify a function to accept an array of numbers and return their sum using the spread syntax. Task 3: Create a deep clone of an object using JSON methods.
Task 4: Write a function that returns a new object, merging two provided objects using the spread syntax. Task 5: Serialize a JavaScript object into a JSON string and then parse it back into an object.

```
function add(n1,n2)
{
    return n1+n2;
}
```

```
console.log(add(1,2));
console.log(add(1,2,3));

const numbers=[1,2,3,45,5];
console.log(add(...numbers));

const clone={
    name:"Ajay",
    roll:202,
    dept:"EEE",
    side:"fullStack"
}
console.log(clone);
let convertJson=JSON.stringify(clone);
console.log(convertJson);
console.log( typeof (convertJson));

const obj1={
    name:"Ajay",
    dept:"EEE"
}

const obj2={
    Roll:202,
    learning:"MERN"
}

function newObj()
{
    const ob=new Object({...obj1,...obj2})
    console.log(ob);
}
newObj();

const serialize={
    Front:"REACT",
    BACK:"NODE AND EXPRESS",
    DataBase:"MONGO DB"
}

let serObj=JSON.stringify(serialize);
console.log(serObj);
let deserObj=JSON.parse(serObj);
```

717821E202_Ajay S

```
console.log(deserObj);
```

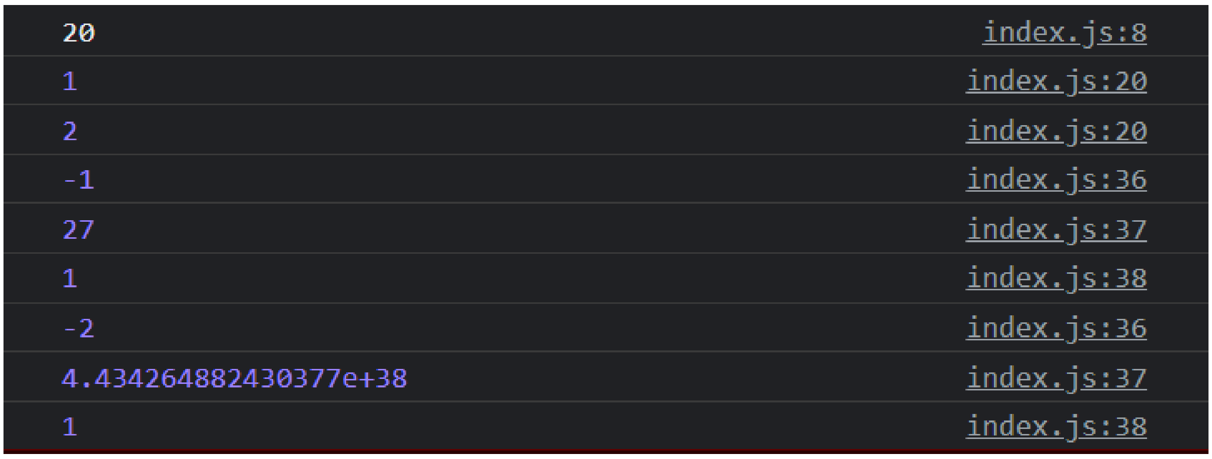


```
3                                                          index.js:7
3                                                          index.js:10
                                                           index.js:18
 ▶ {name: 'Ajay', roll: 202, dept: 'EEE', side: 'fullStack'}
{"name":"Ajay","roll":202,"dept":"EEE","side":"full        index.js:20
Stack"}
string                                                     index.js:21
                                                           index.js:36
 ▶ {name: 'Ajay', dept: 'EEE', Roll: 202, learning: 'MERN'}
{"Front":"REACT","BACK":"NODE AND                          index.js:47
EXPRESS","DataBase":"MONGO DB"}
                                                           index.js:49
 ▶ {Front: 'REACT', BACK: 'NODE AND EXPRESS', DataBase: 'MONGO D
  B'}
```

3. Closure:

```
function add()
{
    let d=10;
    return function()
    {
        console.log(`${d+10}`);
    }
}
let a=add();
a();

function one()
{
    let count=0;
    return function()
    {
        count++;
        console.log(count);
    }
}

let store=one();
store();
store();
```

```
function multiple()
{
    let c1=0,c2=3,c3=1;
    return function()
    {
        c1--;
        c2=c2**c2;
        c3=c3*c3;
        console.log(c1);
        console.log(c2);
        console.log(c3);
    }
}
let value=multiple();
value();
value();

function pri()
{
    let priVariable=0;
    return function()
    {

    }
}
```



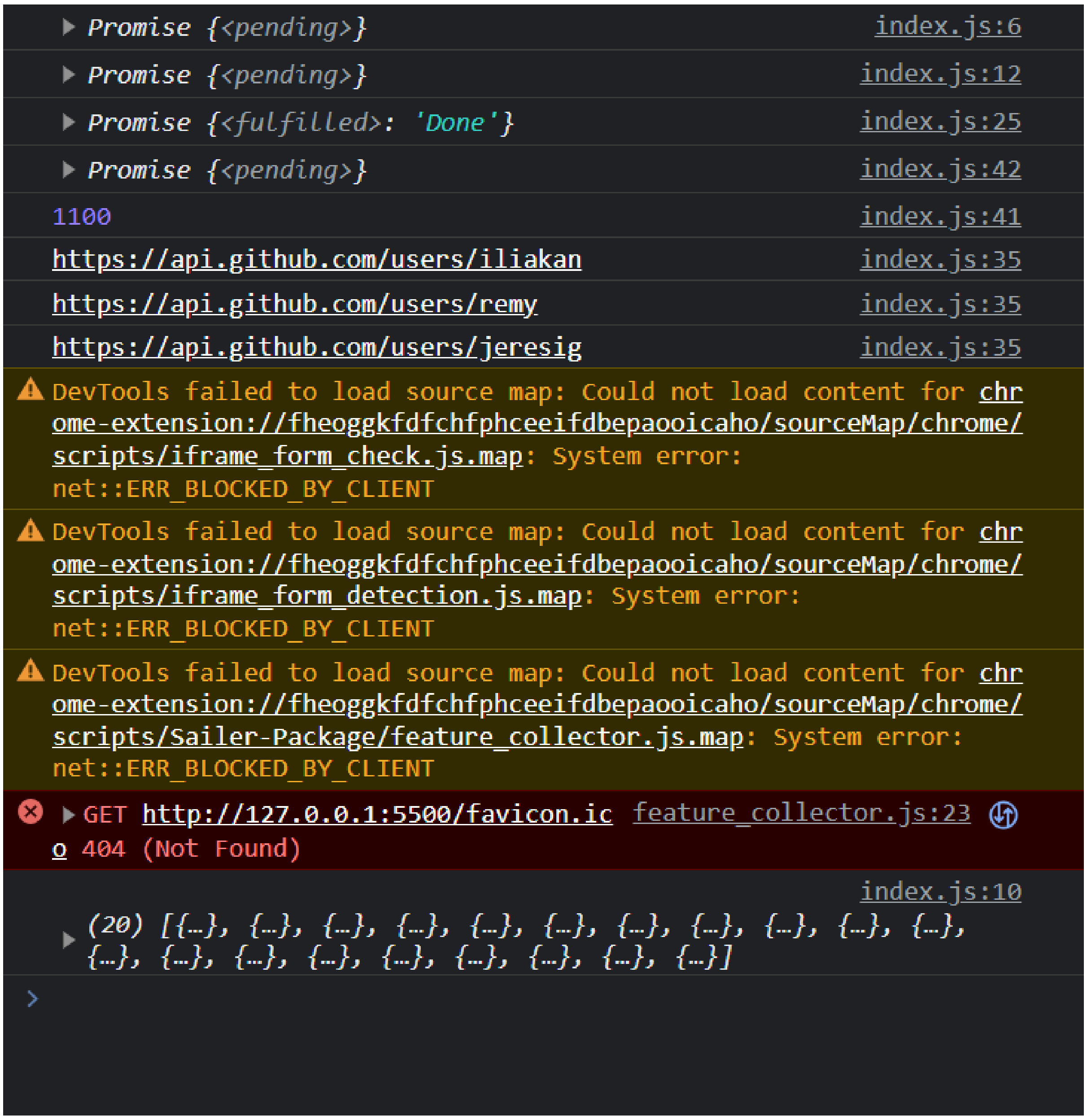
```
20                                   index.js:8
1                                    index.js:20
2                                    index.js:20
-1                                   index.js:36
27                                   index.js:37
1                                    index.js:38
-2                                   index.js:36
4.434264882430377e+38                index.js:37
1                                    index.js:38
```

4. Promise, Promises chaining

```
const myPro=new Promise((resolve,reject)=>
{
    setTimeout(()=>resolve("greetings"),3000);
})
```

```
console.log(myPro)

const dataFetch=new Promise((resolve,reject)=>
{

resolve(fetch('https://fakestoreapi.com/products').then(res=>res.json()
.then(json=>console.log(json))))
})
console.log(dataFetch);


const num=new Promise((resolve,reject)=>{
    let value=Math.floor(Math.random()*10);
    if(value%2==0)
    {
        resolve("Done");
    }
    else if(value%2==1){
        reject("Not Done");
    }
})
console.log(num);

let urls = [
    'https://api.github.com/users/iliakan',
    'https://api.github.com/users/remy',
    'https://api.github.com/users/jeresig'
  ];
  let requests = urls.map(url => fetch(url));
  Promise.all(requests)
    .then(responses => responses.forEach(
      response => console.log(response.url)
    ));

    const mulPro=new Promise((resolve,reject)=>
    {
        resolve(1)
    }).then(function(val1){return val1+10}).then(function(val2){return
val2*100}).then((tot)=>console.log(tot));
    console.log(mulPro);
```

717821E202_Ajay S



```
▶ Promise {<pending>}                                       index.js:6
▶ Promise {<pending>}                                       index.js:12
▶ Promise {<fulfilled>: 'Done'}                             index.js:25
▶ Promise {<pending>}                                       index.js:42
1100                                                        index.js:41
https://api.github.com/users/iliakan                        index.js:35
https://api.github.com/users/remy                           index.js:35
https://api.github.com/users/jeresig                        index.js:35
⚠ DevTools failed to load source map: Could not load content for chr
ome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/
scripts/iframe_form_check.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
⚠ DevTools failed to load source map: Could not load content for chr
ome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/
scripts/iframe_form_detection.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
⚠ DevTools failed to load source map: Could not load content for chr
ome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/
scripts/Sailer-Package/feature_collector.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
✖ ▶GET http://127.0.0.1:5500/favicon.ic feature_collector.js:23
o 404 (Not Found)
                                                            index.js:10
▶ (20) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
  {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
>
```

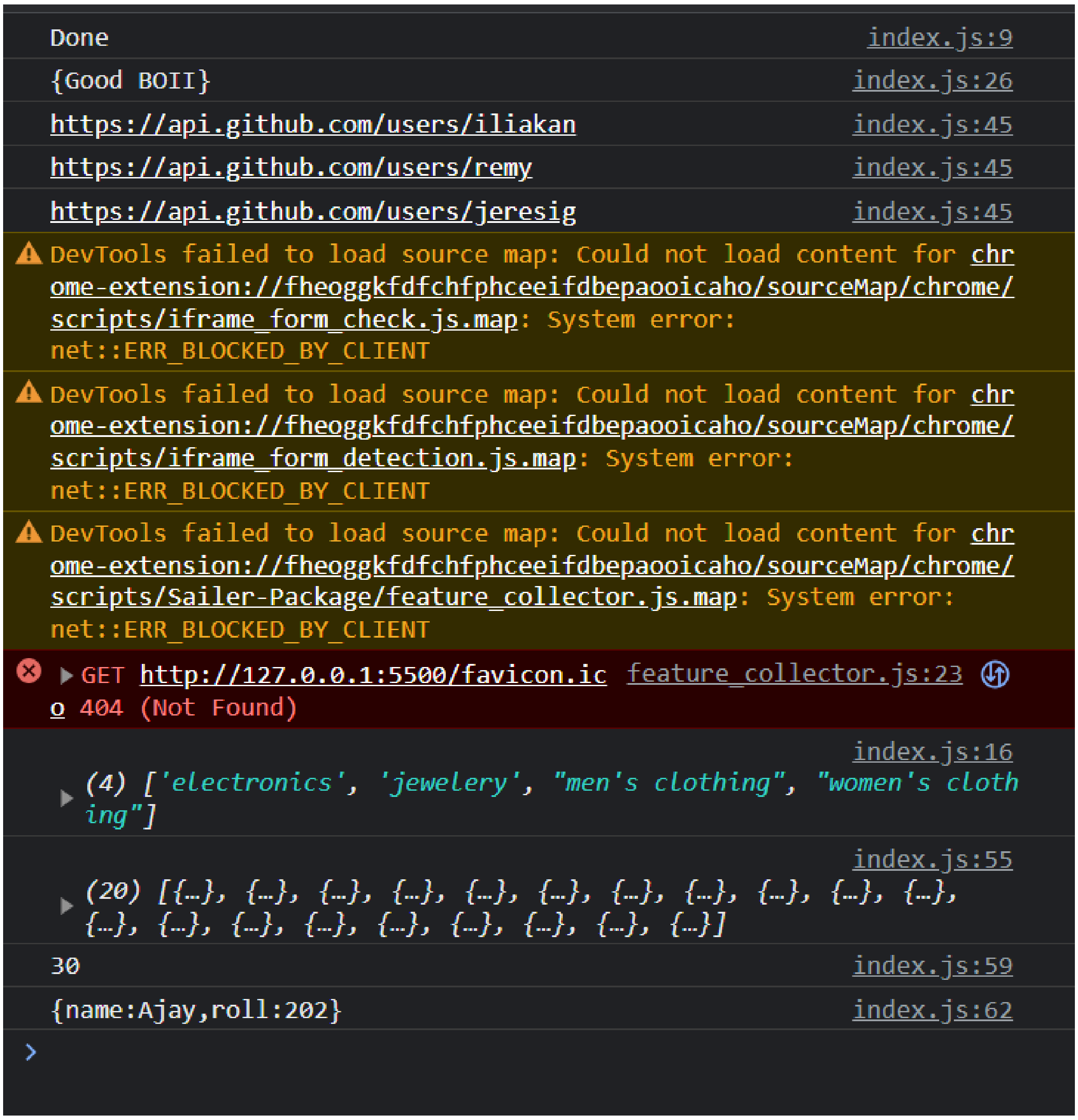
5. Async/await:
Task 1: Rewrite a promise-based function using async/await.
Task 2: Create an async function that fetches data from an API and processes it.
Task 3: Implement error handling in an async function using try/catch.
Task 4: Use async/await in combination with Promise.all.
Task 5: Create an async function that waits for multiple asynchronous operations to complete before proceeding.

```
async function promise()
{
    const pro=new Promise((resolve,reject)=>
    {
        resolve("Done");
    })
    let result=await pro;
    console.log(result);
}
promise();

async function createApi()
{
    let fet= await
fetch('https://fakestoreapi.com/products/categories');
    let insert=await fet.json().then(re=>console.log(re));
}
createApi();

let json="{Good BOII}";
async function Err()
{
    try{
    let pass=JSON.stringify(json);
    let fet=await JSON.parse(pass);
    console.log(fet);
    }catch(err)
    {
        console.log("Something got a Error");
    }
}
Err();


async function run()
{
let urls = [
    'https://api.github.com/users/iliakan',
    'https://api.github.com/users/remy',
    'https://api.github.com/users/jeresig'
  ];
  let requests = urls.map(url => fetch(url));
  let find= await  Promise.all(requests)
  .then(responses => responses.forEach(
    response => console.log(response.url)
  ));;



}
run();
```

717821E202_Ajay S

```
let variable="{name:Ajay,roll:202}"
async  function cat()
{
    let d= await fetch('https://fakestoreapi.com/products');
    let fd=await d.json().then((p)=>console.log(p));
    let num1=10,num2=20;
    let total=num1+num2;
    let tot=await total.toPrecision(2);
    console.log(tot);
    let chan= await JSON.stringify(variable);
    let chaOver=await JSON.parse(chan);
    console.log(chaOver);
}
cat();
```

717821E202_Ajay S



```
Done                                                      index.js:9
{Good BOII}                                               index.js:26
https://api.github.com/users/iliakan                      index.js:45
https://api.github.com/users/remy                         index.js:45
https://api.github.com/users/jeresig                      index.js:45
⚠ DevTools failed to load source map: Could not load content for chr
ome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/
scripts/iframe_form_check.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
⚠ DevTools failed to load source map: Could not load content for chr
ome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/
scripts/iframe_form_detection.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
⚠ DevTools failed to load source map: Could not load content for chr
ome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/
scripts/Sailer-Package/feature_collector.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
⊗ ▶GET http://127.0.0.1:5500/favicon.ic  feature_collector.js:23
o 404 (Not Found)
                                                          index.js:16
 ▶ (4) ['electronics', 'jewelery', "men's clothing", "women's cloth
   ing"]
                                                          index.js:55
 ▶ (20) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
   {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
30                                                        index.js:59
{name:Ajay,roll:202}                                      index.js:62
>
```

7. Browser: DOM Basics

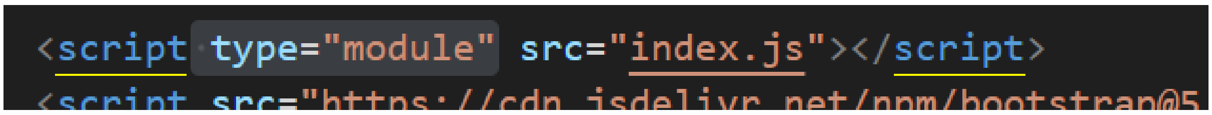
Task 1: Select an HTML element by its ID and change its content using JavaScript.
Task 2: Attach an event listener to a button, making it perform an action when clicked. o
Task 3: Create a new HTML element and append it to the DOM.
Task 4: Implement a function to toggle the visibility of an element.
Task 5: Use the DOM API to retrieve and modify the attributes of an element

```
let change=document.getElementById("change");
change.textContent="hello";
let hi=document.getElementById("hi")
let btn=document.createElement('button');
btn.textContent="Button"
btn.addEventListener('click',perform);
```

```
function perform()
{
     let change=document.getElementById('change');
     change.textContent="hello";
}
let b=document.getElementById('bn');
b.appendChild(btn);

let tog=false;
let btn1=document.createElement('button');
btn1.textContent="Button"
b.appendChild(btn1);
if(tog=!tog)
{
    btn1.addEventListener('click',dis)
    function dis()
    {
        console.log(change.textContent);
        change.classList.toggle("color");
    }
}
```

Modules introduction, Export and Import

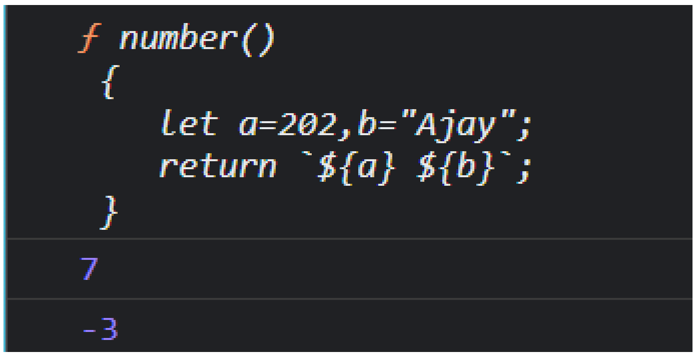
Task 1: Create a module that exports a function, a class, and a variable. Task 2: Import the module in another JavaScript file and use the exported entities. Task 3: Use named exports to export multiple functions from a module. Task 4: Use named imports to import specific functions from a module. Task 5: Use default export and import for a primary function of a module



```
<script type="module" src="index.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5
```

```
export  function number()
 {
    let a=202,b="Ajay";
    return `${a} ${b}`;
 }

    let a=2,b=5
    export let add=a+b;
    export let sub=a-b;
```

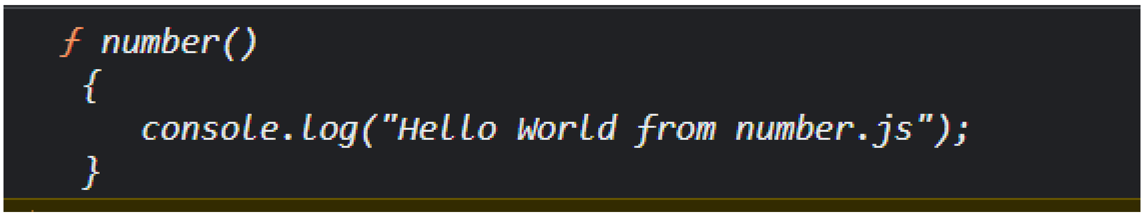
```
 import {number} from './number.js'
 document.body.innerHTML=number;
 console.log(number);

import {add,sub}from './number.js'
document.body.innerHTML=add;
console.log(add);
console.log(sub);
```



```
ƒ number()
 {
    let a=202,b="Ajay";
    return `${a} ${b}`;
 }
7
-3
```

```
export default function number()
 {
    console.log("Hello World from number.js");
 }
```

```
import number from "./number.js";
document.body.innerHTML=number;
console.log(number);
```



```
ƒ number()
 {
    console.log("Hello World from number.js");
 }
```