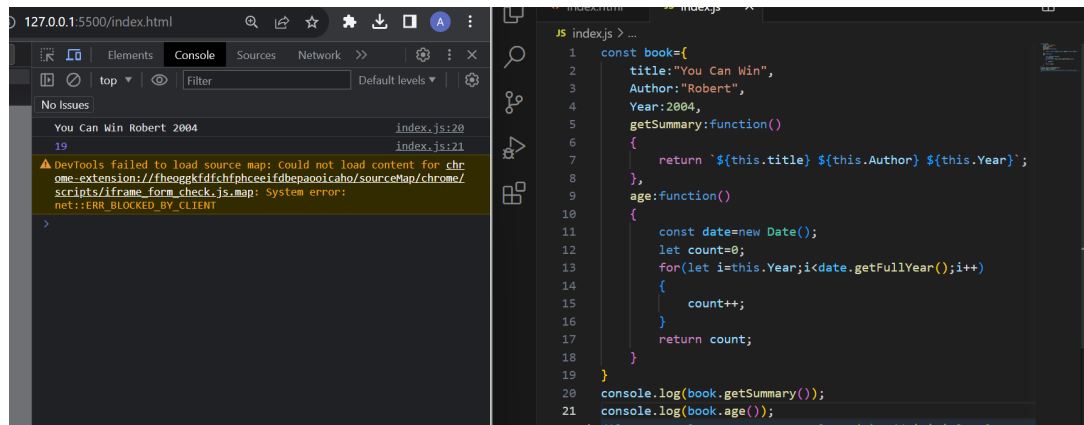


Week 2:

1. Objects and methods, "this":

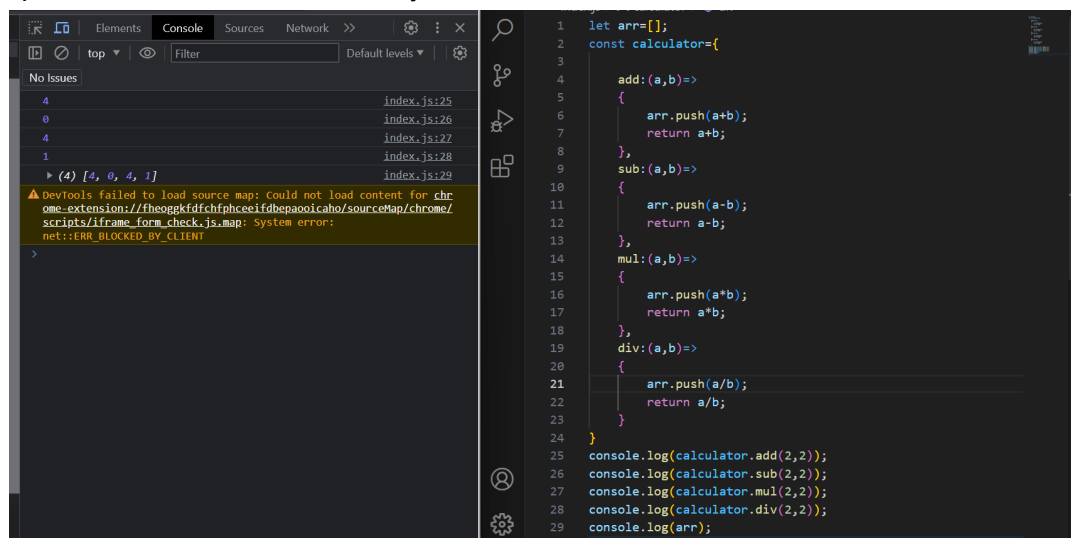
Task 1: Create an object named book with properties: title, author, and yearPublished. Add a method named getSummary that returns a string summarizing the book

Task 2: Modify the book object to include a method named age that calculates how old the book is based on its publication year.



Task 3: Create an object calculator with methods add(), subtract(), multiply(), and divide().

Use the this keyword to refer to the object's properties within these methods. Task 4: Extend the calculator object to include a history array. Each time a calculation is made, store the operation and result in this array



Task 5: Create an object representing a person. Add methods to increase and decrease their age property. Ensure the age doesn't go below 0. Task 6: For the person object, add a method named greet that uses the this keyword to greet with the person's name.

```

1  let count=0;
2  const person={
3    name:"Ajay",
4    inc:()=>
5    {
6      count++;
7      return count;
8    },
9    dec:()=>
10   {
11     if(count>0)
12     {
13       count--;
14       return count;
15     }
16   },
17   greet:()=>
18   {
19     return `This person name is | ${person.name}`
20   }
21 }
22 console.log(person.inc());
23 console.log(person.inc());
24 console.log(person.dec());
25 console.log(person.dec());
26 console.log(person.dec());
27 console.log(person.dec());
28 console.log(person.dec());
29 console.log(person.greet());
30

```

Console output:

```

1  index.js:22
2  index.js:23
1  index.js:24
0  index.js:25
undefined index.js:26
undefined index.js:27
undefined index.js:28
This person name is Ajay index.js:29

```

Error message in console:

```

▲ DevTools failed to load source map: Could not load content for chrome-extension://fhcggkfdhchfphceifdbepaoitaho/sourceMap/chrome-scripts/iframe_form_check.js.map: System error: net::ERR_BLOCKED_BY_CLIENT

```

Task 7: Design a circle object with properties radius and a method to calculate its area using this.radius. Task 8: Extend the circle object with methods to calculate its diameter and circumference

```

1
2  const circle={
3    radius:2,
4    area:function()
5    {
6      let a=2*Math.PI*this.radius;
7      return a.toFixed(3);
8    },
9    diameter:function()
10   {
11     return 2*this.radius;
12   },
13   circumference:function()
14   {
15     return this.radius;
16   }
17 }
18 var console = Console
19 console.log(circle.area());
20 console.log(circle.diameter());
21 console.log(circle.circumference());
22

```

Console output:

```

12.6 index.js:18
4 index.js:19
2 index.js:20

```

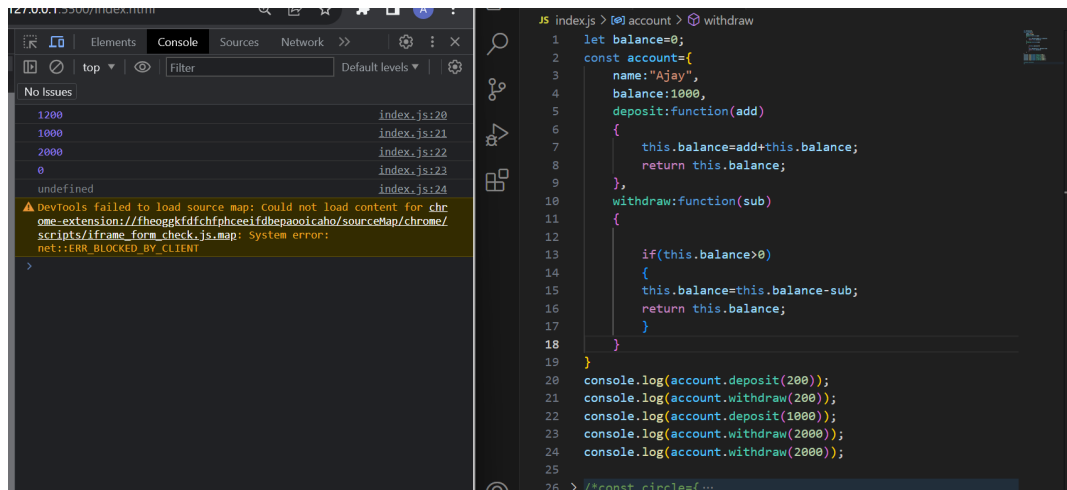
Error message in console:

```

▲ DevTools failed to load source map: Could not load content for chrome-extension://fhcggkfdhchfphceifdbepaoitaho/sourceMap/chrome-scripts/iframe_form_check.js.map: System error: net::ERR_BLOCKED_BY_CLIENT

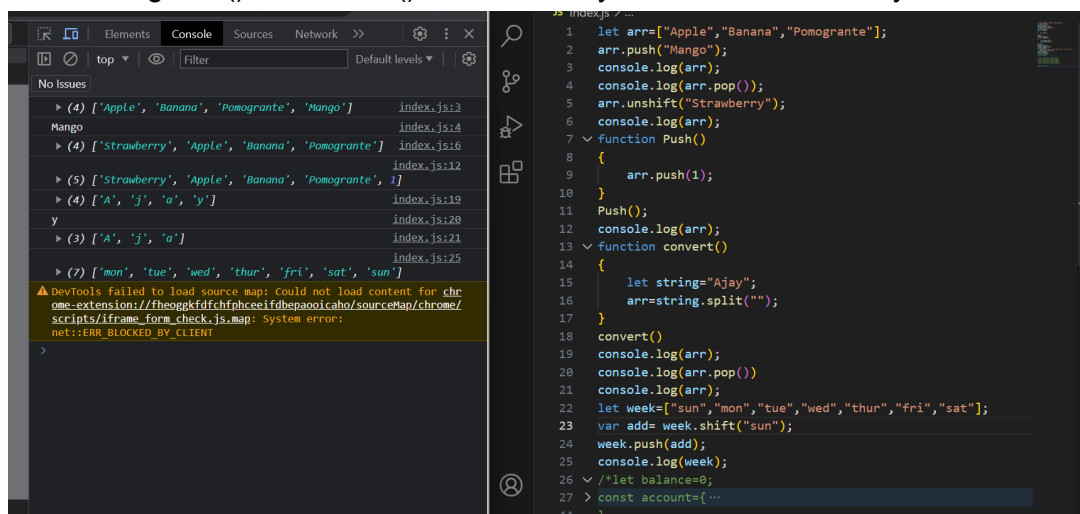
```

Task 9: Create an object account with properties: name, balance and methods: deposit, withdraw. Use the this keyword appropriately. Task 10: For the account object, ensure that the balance can't go negative using the this keyword.

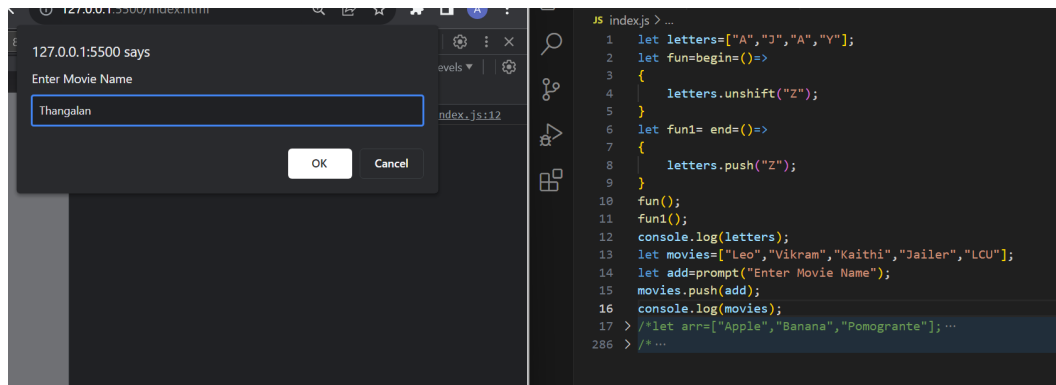


2. Arrays, Array methods (push, pop, shift, unshift)

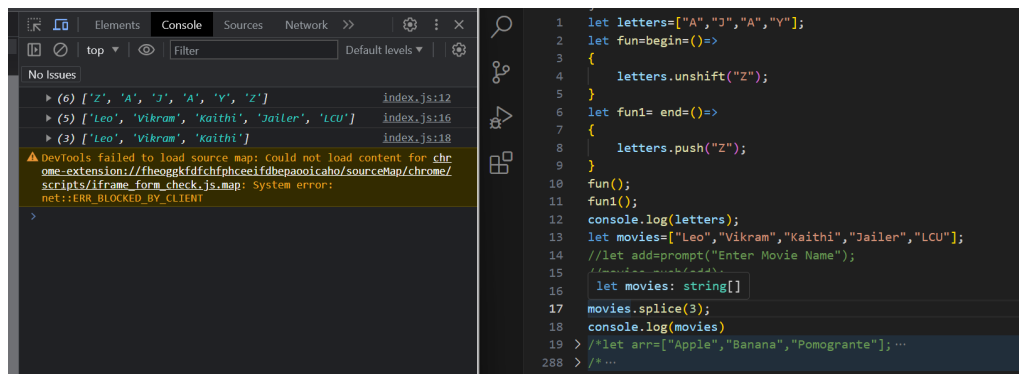
Task 1: Initialize an array of your favorite fruits. Add “Mango” to the end of the array using push(). Task 2: Remove the last fruit from the array using pop(). Task 3: Add “Strawberry” to the beginning of the fruits array using unshift(). Task 4: Remove the first fruit from the array using shift(). Task 5: Create a function that accepts an array of numbers and uses push() to add the number 7 to it. Task 6: Write a function that accepts a string. Convert the string to an array of words and remove the last word using pop(). Task 7: Create an array of days of the week. Using shift() and unshift(), move Sunday to the end of the array.



Task 8: Given an array of letters, write a function that adds a letter ‘Z’ at the beginning and end of the array. Task 9: Initialize an array with five movie names. Ask the user for another movie name and add it to the end of the array.



Task 10: Remove the third item from the movie array



3. Additional methods: map(), filter(), reduce(), slice(), splice()

Task 1: Create an array of numbers. Use map() to create a new array with each number squared. Task 2: Use filter() on an array of numbers to get a new array with only even numbers. Task 3: Create an array of product prices. Use reduce() to find the total price. Task 4: For an array of strings, use map() to create a new array that contains the length of each string. Task 5: Use splice() to remove the third item of an array and replace it with the string "replaced!". Task 6: For an array of integers, use slice() to get a new array containing the 2nd, 3rd, and 4th elements. Task 7: Create an array of names. Use filter() to produce a new array that contains names starting with the letter 'A'.

The screenshot shows the Chrome DevTools interface. The left pane displays the 'Console' tab with a list of log messages. The right pane shows the 'Sources' tab with a JavaScript file named 'index.js' containing the following code:

```

1  const num=[1,2,3,4,5];
2  const sqnum=num.map((num)=>num*num);
3  console.log(sqnum);
4
5
6  const number=[1,2,3,4,5,5,6,7,7];
7  const evenNo=number.filter((num)=>num%2==0);
8  console.log(evenNo);
9
10 const product=[10,90,38];
11 const totalPrice=product.reduce((accumulator,currentValue)=>a
12 console.log(totalPrice);
13
14 const fruits=["apple","banana","cherry","pomogranate"];
15 fruits.splice(2,1,"replaced!");
16 console.log(fruits);
17
18 const integers=[1,2,3,4,5];
19 const done=integers.slice(1,4);
20 console.log(done);
21
22 const names=["Ajay","Kumar","Khan","Ajju","Bhaii"];
23 const Aname=names.filter((char)=>char.charAt(0)=='A');
24 console.log(Aname);
25
26 const score=[90,80,100];
27 const select=score.filter((mark)=>
28 {

```

The console output shows the following log messages:

- (5) [1, 4, 9, 16, 25]
- (3) [2, 4, 6]
- (4) ['apple', 'banana', 'replaced!', 'pomogranate']
- (3) [2, 3, 4]
- (2) ['Ajay', 'Ajju']
- A
- B
- A
- 19.5
- (6) ['car', 'bike', 'auto', 'cycle', 'aeroplane', 'jet']

A DevTools error message is visible at the bottom of the console: "DevTools failed to load source map: Could not load content for chrome-extension://fheoggkfdfchfphceiifdbepaioicaho/sourceMap/chrome/scripts/iframe_form_check.js.map: System error: net::ERR_BLOCKED_BY_CLIENT".

Task 8: For an array of scores (out of 100), use map() to grade each score (e.g., 90-100 = 'A', 80-89 = 'B'). Task 9: Given an array of ages, use reduce() to find the average age. Task 10: Use splice() to insert two new fruits after the second fruit in an array of fruits.

The screenshot shows the Chrome DevTools interface. The left pane displays the 'Console' tab with a list of log messages. The right pane shows the 'Sources' tab with a JavaScript file named 'index.js' containing the following code:

```

29  if(mark>=90 && mark<=100)
30  {
31    console.log("A");
32  }else if(mark>=80 && mark<90)
33  {
34    console.log("B");
35  }
36  })
37  const ages=[19,20,21,18];
38  const avg=ages.length;
39  const avgAge=ages.reduce((accumulator,currentValue)=>accumulat
40  console.log(avgAge/avg);
41
42
43  const fr=["car","bike","aeroplane","jet"];
44  fr.splice(2,0,"auto","cycle");
45  console.log(fr);
46  > /*let letters=["A","J","A","Y"];...
333 > /*...

```

The console output shows the following log messages:

- (5) [1, 4, 9, 16, 25]
- (3) [2, 4, 6]
- (4) ['apple', 'banana', 'replaced!', 'pomogranate']
- (3) [2, 3, 4]
- (2) ['Ajay', 'Ajju']
- A
- B
- A
- 19.5
- (6) ['car', 'bike', 'auto', 'cycle', 'aeroplane', 'jet']

A DevTools error message is visible at the bottom of the console: "DevTools failed to load source map: Could not load content for chrome-extension://fheoggkfdfchfphceiifdbepaioicaho/sourceMap/chrome/scripts/iframe_form_check.js.map: System error: net::ERR_BLOCKED_BY_CLIENT".

Mini Project: "Personal Library"

1. Define the Book Object with methods

```

//Create a Book Object
const Book={
  title:"You Can Win",
  author:"Robert",
  year:2004,
  readStatus:"yes",
  toggleRead:function()
  {
    return `Nice One!`
  },
  toggleReadStatus:function()
  {
    if(this.readStatus=="yes")

```

```
        {
            return "no";
        }
        else{
            return "yes";
        }
    }
}

const Book1={
    title:"Meg",
    author:"Willaim",
    year:2003,
    readStatus:"no",
    toggleRead:function()
    {
        return `Not Bad!`
    },
    toggleReadStatus:function()
    {
        if(this.readStatus=="yes")
        {
            return "no";
        }
        else{
            return "yes";
        }
    }
}

const Book2={
    title:"Veeran",
    author:"Henry",
    year:2002,
    readStatus:"no",
    toggleRead:function()
    {
        return `Not Bad!`
    },
    toggleReadStatus:function()
    {
        if(this.readStatus=="yes")
        {
            return "no";
        }
    }
}
```

```

        else{
            return "yes";
        }
    }
}

const Book4={
    title:"Mass",
    author:"Khan",
    year:2004,
    readStatus:"yes",
    toggleRead:function()
    {
        return `Not Bad!`
    },
    toggleReadStatus:function()
    {
        if(this.readStatus=="yes")
        {
            return "no";
        }
        else{
            return "yes";
        }
    }
}

```

2. Books Collection (Array)

```

//Initailixe Array Lib
const Library=[];
Library.push(Book);
Library.push(Book1);
Library.push(Book2);
Library.push(Book4);

```

3. Array Methods:

```

//Array Methods
//addBook
function addBook(book)
{
    Library.push(book);
}
addBook("Escape");

```

```

addBook("Thriller");

//removeLast
function removeLastBook()
{
    Library.pop();
}
removeLastBook();

//addBookFront
function addBookToFront(book)
{
    Library.unshift(book);
}
addBookToFront("Game Of Thrones");
addBookToFront("Money Hiest");

//removeFirst
function removeFirstBook()
{
    Library.shift();
}
removeFirstBook();
console.log(Library);

```

4. Additional Methods:

```

//Additional Methods

function getAllTitles()
{
    Library.map((e)=>
    {
        console.log(`The Titles of the book is ${e.title}`);
    });
}
getAllTitles();

function getBooksByAuthor(a)
{
    const getAut=Library.filter((val)=>val.author==a);
    console.log(getAut);
}

```



```
getBooksByAuthor("Robert");

function getTotalBooksPublished(value)
{
    const
getYear=Library.filter((y)=>y.year<value).reduce((acc,cuu)=>acc+1,0);
    console.log(getYear);
}
getTotalBooksPublished(2002);

function removeBookByTitle(tit)
{
    let ind=Library.findIndex((index)=>
    {
        if(index.title==tit)
        {
            return true;
        }
    });
    const rem=Library.splice(ind,1);
    console.log(rem);
}
removeBookByTitle("Meg");

function getBookByReadStatus(s)
{
    const ad=Library.filter((y)=>y.readStatus==s)
    console.log(ad);
}
getBookByReadStatus("no");

function getSubLibrary(start,end)
{
    const subli=Library.slice(start,end);
    console.log(subli);
}
getSubLibrary(0,3)
```

717821E202_AJAY S

OUTPUT:

