TASK 1

1.1)

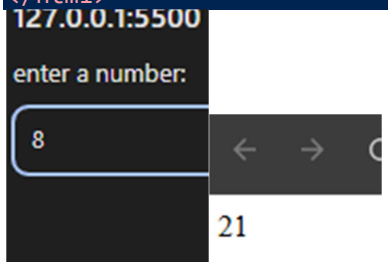
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
    factorial=(n)=>{
        if(n==0){
            return 1;
        }
        else{
            return (n* factorial(n-1))
        }
    }
    let x=prompt("enter a number:");
    document.writeln(factorial(x));
    </script>
</body>
</html>
```

120

1.2)

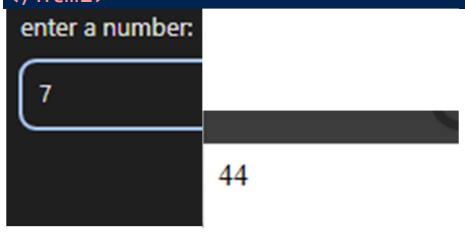
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        fibonacci=(n)=>{
            if(n<=2){
                return 1;
            }
            else{
                return fibonacci(n-1)+fibonacci(n-2);
        }
        }
        let x=prompt("enter a number:");
        document.writeln(fibonacci(x));
```

```
    </script>
</body>
</html>
```



1.3)

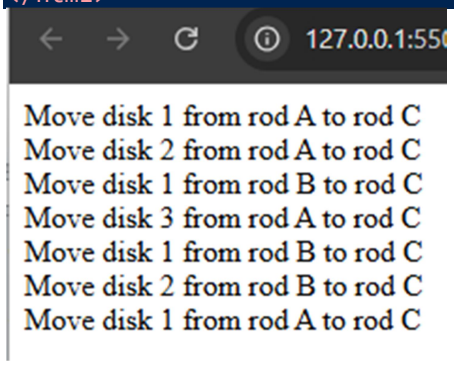
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
    stairs=(n)=>{
        if(n==0){
            return 1;
        }if(n<0){
            return 0;
        }
        else{
            return stairs(n-1)+stairs(n-2)+stairs(n-3);
        }
    }
    let x=prompt("enter a number:");
    document.writeln(stairs(x));
    </script>
</body>
</html>
```



1.5)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        toh=(n,f_rod,t_rod,a_rod)=>{
            if(n==0){
                return;
            }
            toh(n-1,f_rod,t_rod,a_rod)
            document.writeln("Move disk "+ n +" from rod "+ f_rod +" to rod "+
t_rod +"<br>");
            toh(n-1,a_rod,t_rod,f_rod)
        }
        var t=3;
        toh(t,'A','C','B');
    </script>
</body>
</html>
```

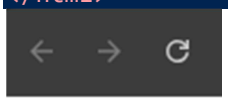


TASK 2

2.1)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
    sum=(...arguments)=>{
        let s=0;
        for(let i=0;i<arguments.length;i++){
            s+=arguments[i]
        }
        return s;
```

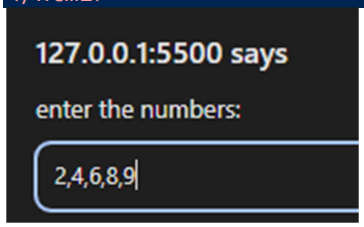
```
    }
    document.writeln(sum(1,2,3,3,4,5,6,7));

</script>

</body>
</html>
```

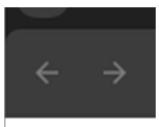


31

2.2)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        function sum(...arguments){
            let total=0;
            for(let i=0;i<arguments.length;i++){
                let n =parseFloat(arguments[i]);
                total+=n;
            }
            return total;
        }
        let input=prompt("enter the numbers:");
        let array=input.split(",");
        document.writeln(sum(...array));
    </script>



</body>
</html>
```





29

2.3)

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        let college={
            name:"bbb",
            dept:"cyber",
            gender:"female"
        };
        document.writeln(typeof(college)+"<br>")
        document.writeln("name :"+college.name +", Department :"
+college.dept+" , Gender :"+college.gender+"<br>");
        let colstg=JSON.stringify(college);
        document.writeln(typeof(colstg)+"<br>")
        document.writeln(colstg+"<br>");
        let colObj=JSON.parse(colstg)
        document.writeln(typeof(colObj)+"<br>");
        document.writeln(`name : ${colObj.name} , Department : ${colObj.dept}
, Gender : ${colObj.gender}`
        );

    </script>
</body>
</html>
```

object
name :bbb, Department :cyber , Gender :female
string
{"name":"bbb","dept":"cyber","gender":"female"}
object
name : bbb , Department : cyber , Gender : female

2.4

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 2.4</title>
</head>
<body>
    <script>
```

```
        function merge(obj1,obj2){
            return{...obj1, ...obj2};
        }
        let stud={
            name:"saha",
            dept:"cyber",
            cgpa:8.0
        };
        let home={
            district:"cuddalore",
            residence:"hosteller"
        };
        let combined=merge(stud,home);
         document.writeln(combined +"<br>");
         document.writeln(JSON.stringify(combined));
    </script>
</body>
</html>
```

```
[object Object]
{"name":"saha","dept":"cyber","cgpa":8,"district":"cuddalore","residence":"hosteller"}
```

2.5

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 2.5</title>
</head>
<body>
    <script>
        function merge(obj1,obj2){
            return{...obj1, ...obj2};
        }
        let stud={
            name:"saha",
            dept:"cyber",
            cgpa:8.0
        };
        let serialize =JSON.stringify(stud);
         document.writeln(serialize+"<br>");

    </script>
</body>
</html>
```

{"name":"saha","dept":"cyber","cgpa":8}

TASK 3

3.1

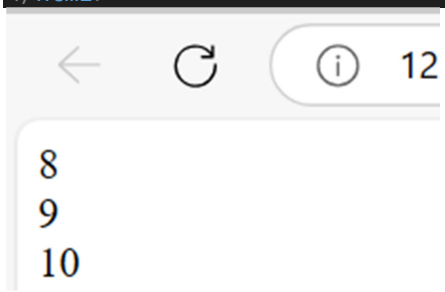
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 3.1</title>
</head>
<body>
    <script>
        function func1(){
            let var1=4;
            return var1
            func2();

        }
        function func2(){
            let var2=func1();
            document.writeln(var2)

        }
        func2();

    </script>
</body>
</html>
```

4

3.2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <Script>
```

```
        function cnt(){
            let s=7;
             return function(){
                s++;
                 return s;
                };
            }
        let a= cnt();
        document.write(a()+"<br>");
        document.write(a()+"<br>");
        document.write(a());
    </Script>
</body>
</html>
```



3.3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        function counter(){
            let s=0;
            return function(){
                s++;
                return s;
            }
        }
        function counter1(){
            let s=0;
            return function(){
                s++;
                return s;
            }
        }
        function counter2(){
```

```
            let s=0;
            return function(){
                s++;
                return s;
            }
        }
        function counter3(){
            let s=0;
            return function(){
                s++;
                return s;
            }
        }
        let c= counter();
        let c1= counter1();
        let c2= counter2();
        let c3= counter3();
        document.write(c()+"<br>");
        document.write(c()+"<br>");
        document.write(c1()+"<br>");
        document.write(c1()+"<br>");
        document.write(c2()+"<br>");
        document.write(c2()+"<br>");
        document.write(c1()+"<br>");
        document.write(c1()+"<br>");
        document.write(c1()+"<br>");
        document.write(c3()+"<br>");
        document.write(c());



    </script>
</body>
</html>
```

1
2
1
2
1
2
3
4
5
1
3

3.4

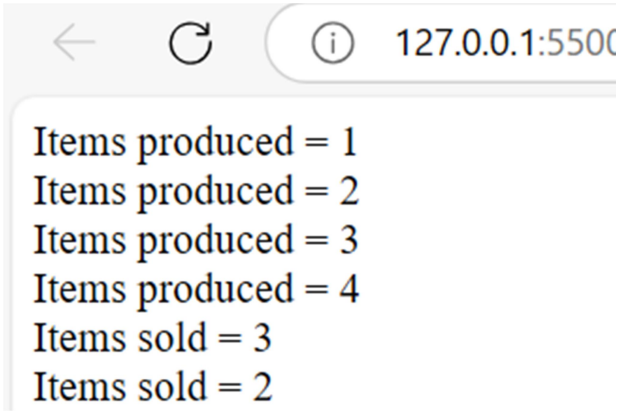
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        function c1(){
            let cnt=2;
            return{
                inc:function(){
                    cnt++;
                    return cnt;
                },
                dec:function(){
                    cnt--;
                    return cnt;
                },
                getcnt:function(){
                    return cnt;
                }
            }
        }
        var counter=c1();
        document.writeln(counter.inc()+"<br>");
        document.writeln(counter.dec()+"<br>");
```

```
        document.writeln(counter.getcnt()+"<br>");
        document.write(counter.cnt);
    </script>
</body>
</html>
```

3
2
2
undefined

3.5

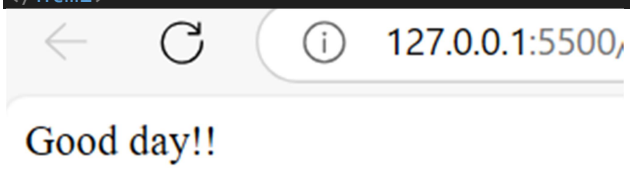
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        function factory(){
            let items=0;
            return{
                production:function(){
                    items++;
                    return items;
                },
                sold:function(){
                    items--;
                    return items;
                }
            }
        }
        let p=factory();
        document.writeln("Items produced = "+p.production()+"<br>")
        document.writeln("Items produced = "+p.production()+"<br>")
        document.writeln("Items produced = "+p.production()+"<br>")
        document.writeln("Items produced = "+p.production()+"<br>")
        document.writeln("Items sold = "+p.sold()+"<br>")
        document.writeln("Items sold = "+p.sold()+"<br>")
    </script>
</body>
</html>
```



TASK 4

4.1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        var greetings=new Promise((resolve,reject)=>{
            setTimeout(()=>resolve("Good day!!"),1000);
        });
        greetings.then((greet)=>{
            document.write(greet);
            })
    </script>
</body>
</html>
```



4.2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 4.2</title>
</head>
```

```
<body>
    <script>
        fetch('https://jsonplaceholder.typicode.com/todos/1')
         .then(Response=>Response.json())
         .then(retrievedData=>{
            console.log('Original Data : ',retrievedData);
            let modifiedData={...retrievedData,transformed:true};
            console.log("updated data : ",modifiedData);
            return retrievedData;
        })
        .catch(fetchError=>console.error('An Error Occurred : ',fetchError));
    </script>
</body>
</html>
```

```
Original Data :                                          task4.2.html:13
▶ {userId: 1, id: 1, title: 'delectus aut autem', completed: false}
updated data :                                           task4.2.html:15
▶ {userId: 1, id: 1, title: 'delectus aut autem', completed: false, transforme
  d: true}
```

4.3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        function randomPromise(){
    return new Promise((resolve,reject) => {
        const randomNum=Math.random();
        console.log(randomNum)
        if (randomNum>0.4) {
            resolve("Success!");
        }else{
            reject("Failure!");
        }
    });
}

randomPromise()
    .then(result=>{
        console.log("Resolved : ",result);
    })
    .catch(error => {
        console.error("Rejected : ",error);
```

```
    });
    </script>
</body>
</html>
```

```
0.6313152293255926                                    task 4.3.html:13
Resolved :  Success!                                  task 4.3.html:24
>
```

4.4

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        const urls = [
            'https://jsonplaceholder.typicode.com/posts/1',
            'https://jsonplaceholder.typicode.com/posts/2',
            'https://jsonplaceholder.typicode.com/posts/3'
        ];
        Promise.all(
            urls.map(url =>
                fetch(url)
                    .then(response => {
                        if (!response.ok) {
                            throw new Error(`Failed to fetch ${url}:
${response.statusText}`);
                        }
                        return response.json();
                    })
                    .catch(error => {
                        console.error(`Error fetching ${url}:`, error);
                        return null;
                    })
            )
        )
        .then(data => {
            console.log('Fetched Data:');
            data.forEach((item, index) => {
                if (item) {
                    console.log(`Resource ${index + 1}:`, item);
                } else {
                    console.log(`Resource ${index + 1}: Fetch failed.`);
                }
```

```
            });
        })
        .catch(error => {
            console.error('Error fetching one or more resources:', error);
        });
    </script>
</body>
</html>
```

```
Fetched Data:                                                    task 4.4.html:31
Resource 1:                                                      task 4.4.html:34
  {userId: 1, id: 1, title: 'sunt aut facere repellat provident occaecati e
▶ xcepturi optio reprehenderit', body: 'quia et suscipit\nsuscipit recusand
  ae consequuntur …strum rerum est autem sunt rem eveniet architecto'}
Resource 2:                                                      task 4.4.html:34
  {userId: 1, id: 2, title: 'qui est esse', body: 'est rerum tempore vitae
▶ \nsequi sint nihil reprehend…aperiam non debitis possimus qui neque nisi
  nulla'}
Resource 3:                                                      task 4.4.html:34
  {userId: 1, id: 3, title: 'ea molestias quasi exercitationem repellat qui
▶ ipsa sit aut', body: 'et iusto sed quo iure\nvoluptatem occaecati omnis
  e…\nmolestiae porro eius odio et labore et velit aut'}
```

4.5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sequential Promises</title>
</head>
<body>
<script>
const taskOne=()=>
    new Promise(resolve=>{
        setTimeout(()=>{
            console.log('Task One completed');
            resolve('Output from Task One');
        },1000);
    });
const taskTwo=previousOutput=>
    new Promise(resolve=>{
        setTimeout(()=>{
            console.log('Task Two completed using : ', previousOutput);
            resolve('Output from Task Two');
        },1000);
    });
const taskThree=previousOutput=>
```

```
    new Promise(resolve=>{
        setTimeout(()=>{
            console.log('Task Three completed using : ', previousOutput);
            resolve('Final Output');
        },1000);
    });
taskOne()
    .then(resultOne=>taskTwo(resultOne))
    .then(resultTwo=>taskThree(resultTwo))
    .then(finalOutput=>{
        console.log('All tasks completed. Final output : ',finalOutput);
    })
    .catch(error=>{
        console.error('An error occurred : ',error);
    });
</script>
</body>
</html>
```

```
Live reload enabled.                                                    task 4.5.html:69
Task One completed                                                      task 4.5.html:13
Task Two completed using :  Output from Task One                        task 4.5.html:20
Task Three completed using :  Output from Task Two                      task 4.5.html:27
All tasks completed. Final output :  Final Output                       task 4.5.html:35
```

TASK 5

5.1

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Task 5.1</title>
</head>
<body>
<script>
async function runTask(){
    try{
        let message=await new Promise((fulfill,reject) => {
            setTimeout(()=>fulfill('Greetings!!'),1000);
        });
        console.log(message);
    } catch(e){
        console.error('error occurred:',e);
```

```
    }
}
runTask();
</script>
</body>
</html>
```

```
Greetings!!                                    task 5.1.html:15
```

5.2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 5.2</title>
</head>
<body>
    <script>
        async function retrieveAndTransform() {
    try {
        const apiResponse = await
fetch('https://jsonplaceholder.typicode.com/posts');
        const rawData = await apiResponse.json();
        const enhancedData = rawData.map(item => ({ ...item, isTransformed:
true }));
        console.log('Enhanced Data:', enhancedData);
    } catch (err) {
        console.error('An error occurred:', err);
    }
}
retrieveAndTransform();

    </script>
</body>
</html>
```

```
Live reload enabled.                                   task 5.2.html:51
Enhanced Data:                                         task 5.2.html:15
▼ Array(100) i
  ▶ 0: {userId: 1, id: 1, title: 'sunt aut facere repellat provident occa
  ▶ 1: {userId: 1, id: 2, title: 'qui est esse', body: 'est rerum tempore
  ▶ 2: {userId: 1, id: 3, title: 'ea molestias quasi exercitationem repel
  ▶ 3: {userId: 1, id: 4, title: 'eum et est occaecati', body: 'ullam et
  ▶ 4: {userId: 1, id: 5, title: 'nesciunt quas odio', body: 'repudiandae
```

5.3

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task5.3</title>
</head>
<body>
    <script>
        async function errorHandling() {
            try {
        const outcome=await new Promise((resolve, reject) => {
            setTimeout(()=>reject(new Error('Error')), 1000);
        });
        console.log(outcome);
    } catch (e){
        console.error('Error encountered:',e.message);
    }
}
errorHandling();
    </script>
</body>
</html>
```

```
Live reload enabled.                                    task 5.3.html:50
▶ Error encountered: Error                              task 5.3.html:17
```

5.4

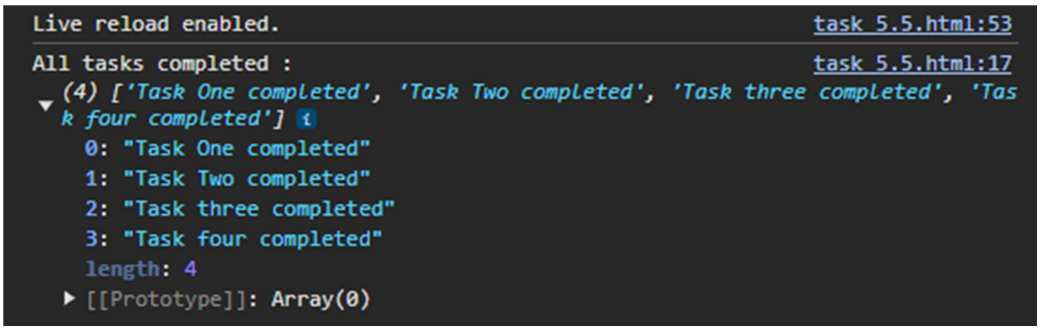
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 5.4</title>
</head>
<body>
    <script>
        async function executeAllTasks() {
            const stepOne=()=>new
Promise(resolve=>setTimeout(()=>resolve('Step One completed'),1000));
            const stepTwo=()=>new
Promise(resolve=>setTimeout(()=>resolve('Step Two completed'),2000));
            try {
                const outcomes = await Promise.all([stepOne(), stepTwo()]);
                console.log('All steps completed : ',outcomes);
            } catch (e) {
        console.error('Error during task execution : ',e);
    }
```

```
        }
        executeAllTasks();

    </script>
</body>
</html>
```

```
All steps completed :                                    task 5.4.html:15
▼ (2) ['Step One completed', 'Step Two completed'] t
    0: "Step One completed"
    1: "Step Two completed"
    length: 2
  ▶ [[Prototype]]: Array(0)
>
```

5.5

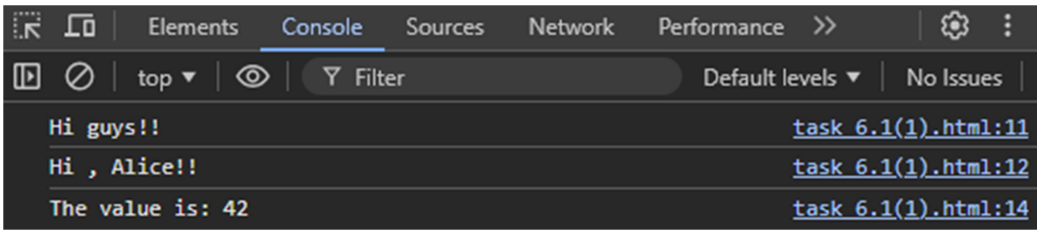
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 5.4</title>
</head>
<body>
    <script>
        async function multipleasync() {
            const task1=()=>new Promise(resolve=>setTimeout(()=>resolve('Task
One completed'),1000));
            const task2=()=>new Promise(resolve=>setTimeout(()=>resolve('Task
Two completed'),2000));
            const task3=()=>new Promise(resolve=>setTimeout(()=>resolve('Task
three completed'),3000));
            const task4=()=>new Promise(resolve=>setTimeout(()=>resolve('Task
four completed'),4000));
            try {
                const outcomes = await
Promise.all([task1(),task2(),task3(),task4()]);
                console.log('All tasks completed : ',outcomes);
            } catch (e) {
        console.error('Error during task execution : ',e);
    }
        }
        multipleasync();

    </script>
</body>
</html>
```



TASK 6

6.1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 6.1</title>
</head>
<body>
    <script type="module">
        import{var1,func,MyClass} from'./demo1.js';
        console.log(var1);
        console.log(func("Alice"));
        const x = new MyClass(42);
        console.log(x.displayValue());

     </script>
</body>
</html>

export let var1 = "Hi guys!!";
export function func(name) {
    return `Hi , ${name}!!`;
}

export class MyClass {
    constructor(value) {
        this.value = value;
    }

    displayValue() {
        return `The value is: ${this.value}`;
    }
}
```



6.2

task 6.2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 6.2</title>
</head>
<body>
    <script type="module" src="demo2.1.js">
     </script>
</body>
</html>
```

demo 1.js

```
export let var1 = "Hi guys!!";
export function func(name) {
    return `Hi , ${name}!!`;
}

export class MyClass {
    constructor(value) {
        this.value = value;
    }

    displayValue() {
        return `The value is: ${this.value}`;
    }
}
```

demo 2.1.js

```
import{var1,func,MyClass} from'./demo1.js';
        console.log(var1);
        console.log(func("Alpha"));
        const x = new MyClass(69);
        console.log(x.displayValue());
```

6.3

Task 6.3.html

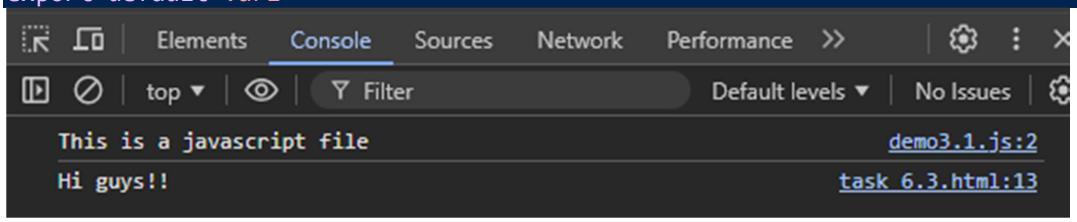
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 6.3</title>
</head>
<body>
    <script type="module">
        import y from '/demo3.1.js';
        y()
        import var1 from '/demo3.2.js';
        console.log(var1);
    </script>
</body>
</html>
```

Demo3.1.js

```
export default function y(){
    console.log("This is a javascript file");
}
```

Demo3.2.js

```
let var1="Hi guys!!";
export default var1
```



6.4

Task 6.4

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 6.4</title>
</head>
<body>
    <script type="module">
```

```
        import var1 from '/demo4.js';
        console.log(var1);
        </script>
</body>
</html>
```

Demo 4

```
let var1 = "Hi guys!!";
export default var1
export function y(){
    console.log("This is a javascript file");
}
```

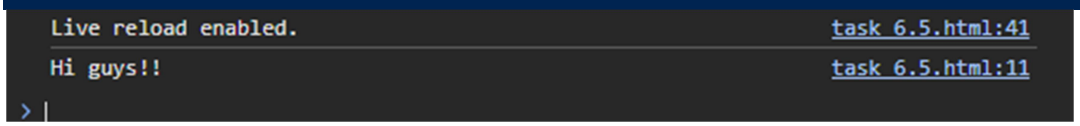
6.5

Task 6.5

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 6.5</title>
</head>
<body>
    <script type="module">
        import var1 from '/demo4.js';
        console.log(var1);
        </script>
</body>
</html>
```

Demo4.js

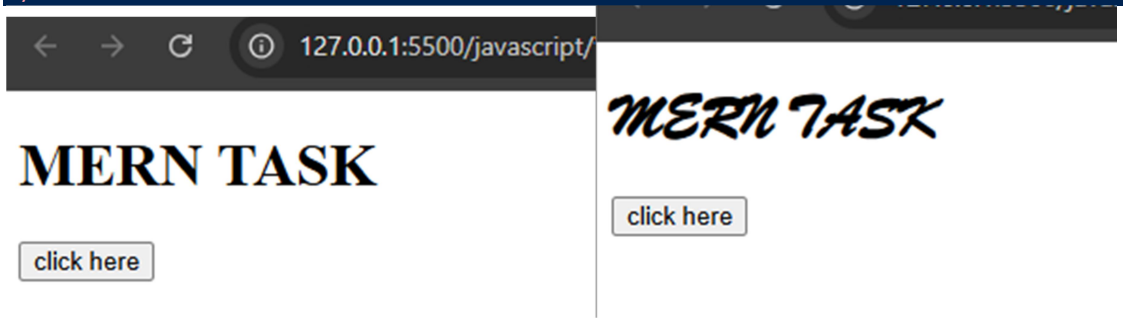
```
let var1 = "Hi guys!!";
export default var1
export function y(){
    console.log("This is a javascript file");
}
```



TASK 7

7.1

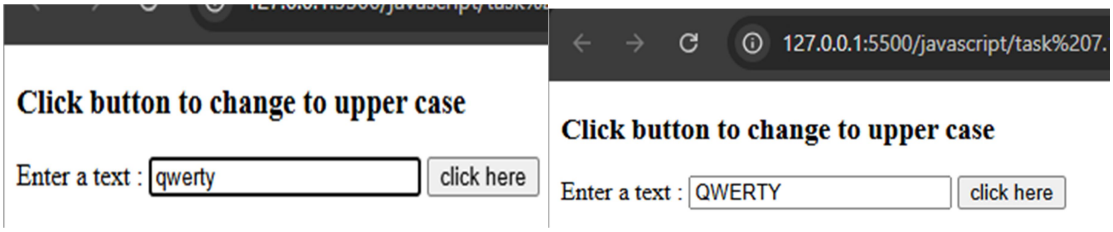
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 7.1</title>
</head>
<body>

    <h1 id="sample">MERN TASK</h1>
    <button onclick="changeColor()">click here</button>

    <script>
        function changeColor(){
        let a=document.getElementById("sample");
        a.style.fontFamily="Brush Script MT";
        }
    </script>
</body>
</html>
```



7.2

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 7.2</title>
</head>
<body>

    <h3>Click button to change to upper case</h3>
    Enter a text : <input type="text" id="sample">
    <button onclick="ChangeToUpperCase()">click here</button>

    <script>
        function ChangeToUpperCase(){
        let a=document.getElementById("sample");
        a.value=a.value.toUpperCase()
```

```
        }
    </script>
</body>
</html>
```



7.3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 7.3</title>
</head>
<body>
    <h2 id="sample">create new element and add to DOM</h2>
    <button onclick="NewElement()">New element</button>
    <script>
        function NewElement(){
            let a=document.createElement("h2")
            a.textContent="This is new element"
            document.body.appendChild(a);
        }
    </script>
</body>
</html>
```

7.4

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>task 7.4</title>
```

```
</head>
<body>
    <h2>Function to toggle the visibility of an element </h2>
    <h3 id="sample"><i> This  my toggled text!!!</i></h3>
    <button onclick="toggle()">click here</button>

    <script>
        function toggle(){
            let a=document.getElementById("sample")
            if(a.style.display==='none'){
                a.style.display="block";
            }else{
                a.style.display='none';
            }
        }
    </script>
</body>
</html>
```

7.5

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 7.5</title>
</head>
<body>
    <h2>Retrieve and modify the attributes</h2>
    <p id="sample" style="font-family : monospace;"><b>this is monospace
font</b></p>
    <button onclick="Modify()">Click here</button>

    <script>
        function Modify(){
            let a=document.getElementById("sample")
            document.writeln("current attribute :
",a.getAttribute("style"),"<br>")
            a.setAttribute("style","font-family : Times New Roman;")
            let b=a.getAttribute("style")
            document.writeln("modified attribute : ",b);


        }
```

```
    </script>
</body>
</html>
```