CLOSURE:

TASK 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 mergeObjects(obj1, obj2) {
     const mergedObject = Object.assign({}, obj1, obj2);
     document.writeln(JSON.stringify(mergedObject));
}
const object 1 = \{ a: 1, b: 2 \};
const object2 = \{ c: 3, d: 4 \};
mergeObjects(object1, object2);
           </script>
</body>
</html>
                                   Ocument
                                                                                                 ighthat is a compared to the c
                                                               G
     {"a":1,"b":2,"c":3,"d":4}
```

TASK 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>
    const student = {
    name: "Madhan",
    age: 20,
    dept: "EEE"
   };
   const jsonString = JSON.stringify(student);
   document.writeln("object to JSON string: " + jsonString + "<br/>');
    const parsedObject = JSON.parse(jsonString);
   document.writeln("JSON string to object: " + JSON.stringify(parsedObject));
  </script>
</body>
</html>
                  ① File C:/Users/Kirut/Desktop/madhan/day.html
            G
object to JSON string: {"name":"Madhan","age":20,"dept":"EEE"}
JSON string to object: {"name":"Madhan","age":20,"dept":"EEE"}
```

PROMISE, PROMISES CHAINING:

TASK 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>
    function greetings(){
    return new Promise(()=>{
     setTimeout(()=>{
     document.writeln("Hello world");
     },2000)
    })
  }
let a=greetings();
</script>
</body>
</html>
                  • File C:/Users/Kirut/Desktop/madhan/day.h
 Good morning
TASK 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>
     fetch('https://jsonplaceholder.typicode.com/users')
 .then(response \Rightarrow {
  if (!response.ok) {
   throw new Error('Failed to fetch data');
  return response.json();
 })
 .then(users => {
  console.log('Fetched users:', users);
  return users.filter(user => user.address.city === 'coimbatore');
 })
 .then(filteredUsers => {
  console.log('Filtered Users from coimbatore:', filteredUsers);
  return filteredUsers.map(user => user.name);
 })
 .then(userNames => \{
  console.log('User Names from coimbatore:', userNames);
 })
 .catch(error => {
  console.error('Error:', error);
 });
  </script>
```

```
</body>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

> Fetched users: (10) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]

> Filtered Users from coimbatore: (0) []

> User Names from coimbatore: (0) []
```

TASK 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 prom=new Promise((resolve,reject)=>{
    const x=2;
    if(x==0) resolve("Success");
    else resolve("Not successful");
    });
    prom.then(result=>{
    document.writeln(result);
    })
    .catch(error=>{
    document.writelnerror(error);
    });
  </script>
</body>
</html>
```



Not successful

TASK 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/todos/1',
'https://jsonplaceholder.typicode.com/todos/2',
'https://jsonplaceholder.typicode.com/todos/3'
];
function fetchData(url) {
return fetch(url)
.then(response => {
if (!response.ok) {
throw new Error('HTTP error! Status: ${response.status}');
}
return response.json();
})
.catch(error => {
```

```
throw error;
});
Promise.all(urls.map(fetchData))
.then(results => {
document.writeln('All resources fetched: <br/>
y', JSON.stringify(results));
})
.catch(error => {
document.writeln('Error fetching resources: <br/>
y', JSON.stringify(error));
});
   </script>
</body>
</html>
 ← → C ① File C:/Users/Kirut/Desktop/madhan/day.html
                                                                                                            ☆
 All resources fetched:
 [{"userId":1,"id":1,"title":"delectus aut autem","completed":false}, {"userId":1,"id":2,"title":"quis ut nam facilis et officia qui","completed":false}, {"userId":1,"id":3,"title":"fugiat veniam minus","completed":false}]
TASK 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 fetchData(data) {
 return new Promise((resolve) => {
   setTimeout(() \Rightarrow \{
```

```
resolve(data);
  }, 1000);
 });
}
fetchData('Step 1: Fetch data')
 .then((result) => {
  console.log(result);
  return fetchData('Step 2: Process data');
 })
 .then((result) => \{
  console.log(result);
  return fetchData('Step 3: Complete task');
 })
 .then((result) => {
  console.log(result);
 })
 .catch((error) => {
  console.error('Error:', error);
 });
  </script>
</body>
</html>
 PROBLEMS
                        DEBUG CONSOLE
   Step 3: Complete task
```