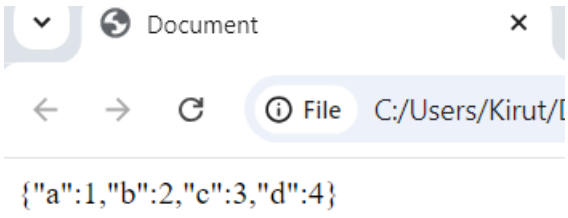
CLOSURE:

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 object1 = { a: 1, b: 2 };
const object2 = { c: 3, d: 4 };

mergeObjects(object1, object2);
    </script>
</body>
</html>
```



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>
```

PROMISE,PROMISES CHAINING:

TASK 1:

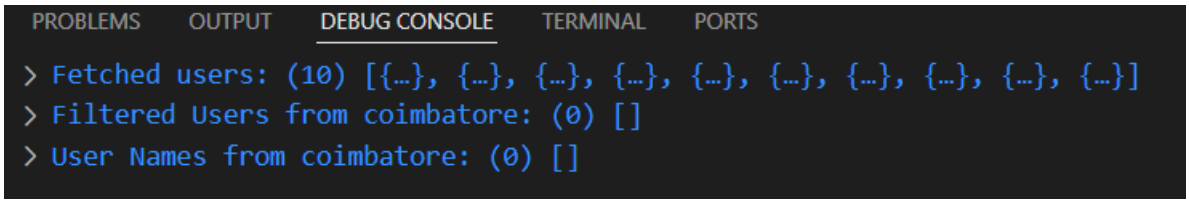
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        function greetings(){
        return new Promise(()=>{
          setTimeout(()=>{
          document.writeln("Hello world");
         },2000)
        })
    }
let a=greetings();
</script>
</body>
</html>
```

TASK 2:

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

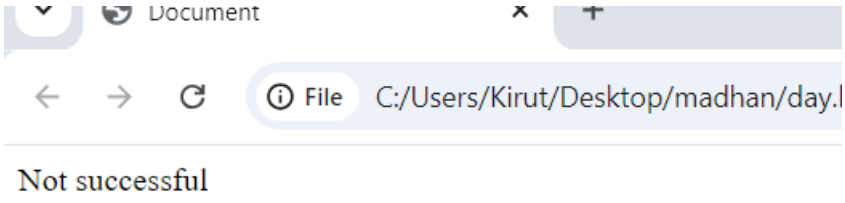
```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        fetch('https://jsonplaceholder.typicode.com/users')
  .then(response => {
    if (!response.ok) {
      throw new Error('Failed 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>
</html>
```



TASK 3:

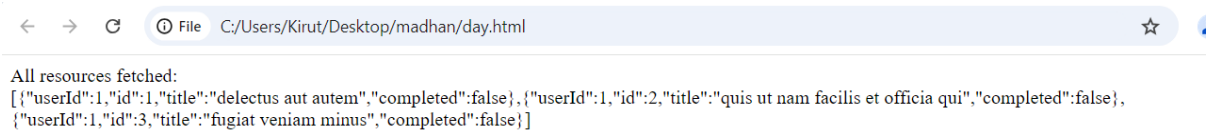
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        let prom=new Promise((resolve,reject)=>{
        const x=2;
        if(x==0) resolve("Success");
        else resolve("Not successful");
        });
        prom.then(result=>{
        document.writeln(result);
        })
        .catch(error=>{
        document.writelnerror(error);
        });
    </script>
</body>
</html>
```



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>', JSON.stringify(results));
})
.catch(error => {
document.writeln('Error fetching resources:<br>', JSON.stringify(error));
});
    </script>
</body>
</html>
```



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(() => {
```

```
      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>
```