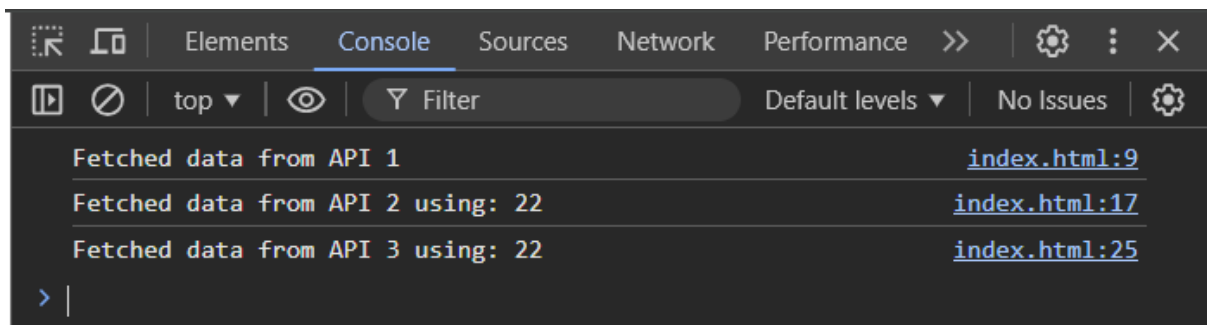


717823E229

TASK – 5.1

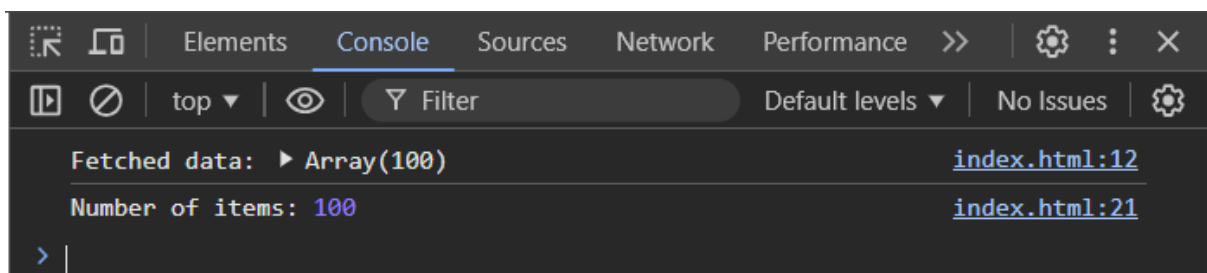
```
<!DOCTYPE html>
<html>
<body>
  <script>
function fetchDataFromAPI1() {

return new Promise((resolve) => {
setTimeout(() => {
console.log("Fetched data from API 1");
resolve("Data from API 1");
}, 1000);
});
}
function fetchDataFromAPI2(data) {
return new Promise((resolve) => {
setTimeout(() => {
console.log(`Fetched data from API 2 using: ${data}`);
resolve("Data from API 2");
}, 1000);
});
}
function fetchDataFromAPI3(data) {
return new Promise((resolve) => {
setTimeout(() => {
console.log(`Fetched data from API 3 using: ${data}`);
resolve("Data from API 3");
}, 1000);
});
}
async function chainPromises() {
let data1=22;
const a=await fetchDataFromAPI1();
const b=await fetchDataFromAPI2(data1);
const c=await fetchDataFromAPI3(data1);
document.writeln(status)
}
chainPromises();
  </script>
</body>
</html>
```



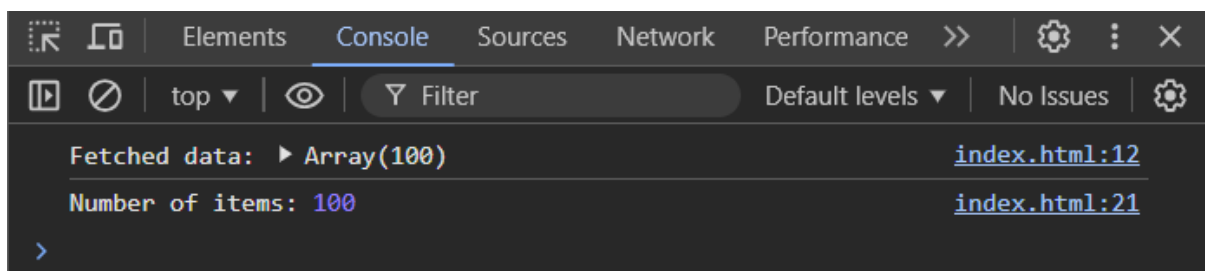
TASK – 5.2

```
<!DOCTYPE html>
<html>
<body>
  <script>
    async function fetchDataAndProcess(url) {
      try {
        const response = await fetch(url);
        if (!response.ok) {
          throw new Error("Network response was not ok");
        }
        const data = await response.json();
        console.log("Fetched data:", data);
        return data.length;
      } catch (error) {
        console.error("Error fetching data:", error);
      }
    }
    const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
    fetchDataAndProcess(apiUrl).then((result) => {
      if (result !== undefined) {
        console.log('Number of items:', result);
      }
    });
  </script>
</body>
</html>
```



TASK – 5.3

```
<!DOCTYPE html>
<html>
<body>
  <script>
    async function fetchDataAndProcess(url) {
      try {
        const response = await fetch(url);
        if (!response.ok) {
          throw new Error("Network response was not ok");
        }
        const data = await response.json();
        console.log("Fetched data:", data);
        return data.length;
      } catch (error) {
        console.error("Error fetching data:", error);
      }
    }
    const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
    fetchDataAndProcess(apiUrl).then((result) => {
      if (result !== undefined) {
        console.log('Number of items:', result);
      }
    });
  </script>
</body>
</html>
```



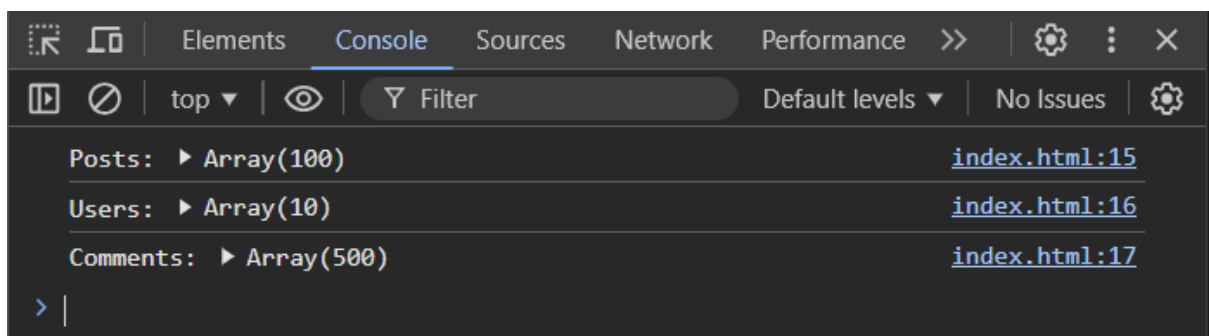
TASK – 5.4

```
<!DOCTYPE html>
<html>
<body>
  <script>
    async function fetchMultipleResources() {
      const urls = [
        'https://jsonplaceholder.typicode.com/posts',
        'https://jsonplaceholder.typicode.com/users',
        'https://jsonplaceholder.typicode.com/comments'
      ];
    }
  </script>
</body>
</html>
```

```

try {
const fetchPromises = urls.map(url => fetch(url).then(response => response.json()));
const results = await Promise.all(fetchPromises);
console.log('Posts:', results[0]);
console.log('Users:', results[1]);
console.log('Comments:', results[2]);
} catch (error) {
console.error('Error fetching data:', error);
}
}
fetchMultipleResources();
</script>
</body>
</html>

```



TASK – 5.5

```

<!DOCTYPE html>
<html>
<body>
  <script>
async function waitForMultipleOperations() {
try {
const operation1 = new Promise((resolve) => setTimeout(() => resolve('Operation 1 Complete'),
2000));
const operation2 = new Promise((resolve) => setTimeout(() => resolve('Operation 2 Complete'),
3000));
const operation3 = new Promise((resolve) => setTimeout(() => resolve('Operation 3 Complete'),
1000));
const results = await Promise.all([operation1, operation2, operation3]);
console.log('All operations completed:');
results.forEach(result => console.log(result));
} catch (error) {

console.error('Error:', error);
}
}
waitForMultipleOperations();

```

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

