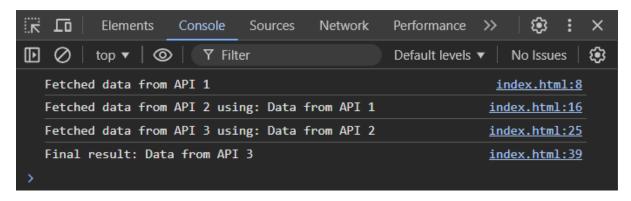
```
717823E229
TASK - 4.1
<!DOCTYPE html>
<html>
<body>
 <script>
function greetAfterSeconds() {
return new Promise((resolve) => {
setTimeout(() => {
resolve("Welcome!");
}, 2000);
});
}
greetAfterSeconds().then((greeting) => {
document.writeln(greeting);
});
 </script>
</body>
</html>
              index.html
                           (i) File C:/Users/kiran/OneDri
                  G
  Welcome!
TASK - 4.2
<!DOCTYPE html>
<html>
<body>
 <script>
function fetchData(url) {
return fetch(url)
.then(response => response.json())
.then(data => {
console.log('Fetched data:', data);
return data;
})
.then(data => {
const count = data.length;
console.log('Number of items:', count);
})
.catch(error => {
console.log('Error:', error);
});
}
```

```
const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
fetchData(apiUrl);
 </script>
</body>
</html>
 Y Filter
                                                          Default levels ▼
                                                                            No Issues
     Fetched data: ▶ Array(100)
                                                                          index.html:9
    Number of items: 100
                                                                         index.html:14
TASK - 4.3
<!DOCTYPE html>
<html>
<body>
 <script>
function randomPromise() {
return new Promise((resolve, reject) => {
const randomNumber = Math.random();
if (randomNumber > 2.5) {
resolve("Success! The random number was greater than 2.5.");
} else {
reject("Failure! The random number was less than or equal to 2.5.");
}
});
}
randomPromise()
.then((message) => {
console.log(message);
})
.catch((error) => {
console.log(error);
});
 </script>
</body>
</html>
 K
                                                                                ₿
    ㅁ
             Elements
                         Console
                                    Sources
                                               Network
                                                          Performance
                                                                        >>
                    0
            top ▼

▼ Filter

                                                          Default levels ▼
                                                                             No Issues
    Failure! The random number was less than or equal to 2.5.
                                                                         index.html:20
TASK - 4.4
<!DOCTYPE html>
<html>
<body>
 <script>
```

```
function fetchDataFromAPI1() {
return new Promise((resolve) => {
setTimeout(() => {
console.log("Fetched data from API 1");
resolve("Data from API 1");
}, 2000);
});
}
function fetchDataFromAPI2(data) {
return new Promise((resolve) => {
setTimeout(() => {
console.log(`Fetched data from API 2 using: ${data}`);
resolve("Data from API 2");
}, 2000);
});
}
function fetchDataFromAPI3(data) {
return new Promise((resolve) => {
setTimeout(() => {
console.log(`Fetched data from API 3 using: ${data}`);
resolve("Data from API 3");
}, 2000);
});
}
function chainPromises() {
fetchDataFromAPI1()
.then((data1) => {
return fetchDataFromAPI2(data1);
})
.then((data2) => {
return fetchDataFromAPI3(data2);
})
.then((data3) => {
console.log(`Final result: ${data3}`);
})
.catch((error) => {
console.error("Error:", error);})
chainPromises();
 </script>
</body>
</html>
```



```
TASK - 4.5
```

```
<!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);
});
}
function chainPromises() {
fetchDataFromAPI1()
.then((data1) => {
return fetchDataFromAPI2(data1);
})
.then((data2) => {
return fetchDataFromAPI3(data2);
```

```
})
.then((data3) => {
console.log(`Final result: ${data3}`);
.catch((error) => {
console.error("Error:", error);})
}
chainPromises();
 </script>
</body>
</html>
 K [0
                                                                              €3
             Elements
                         Console
                                   Sources
                                              Network
                                                         Performance
         top ▼ | ③ | (
                            Y Filter
                                                                           No Issues
                                                         Default levels ▼
    Fetched data from API 1
                                                                         index.html:8
    Fetched data from API 2 using: Data from API 1
                                                                       index.html:16
    Fetched data from API 3 using: Data from API 2
                                                                       index.html:25
    Final result: Data from API 3
                                                                       index.html:39
```