Lesson 05 Demo 04

Working with Promises

Objective: To implement promises and asynchronous functions in JavaScript, focusing on mastering asynchronous control flow and error handling to enhance functionality and reliability within a web development framework

Tools required: Visual Studio Code

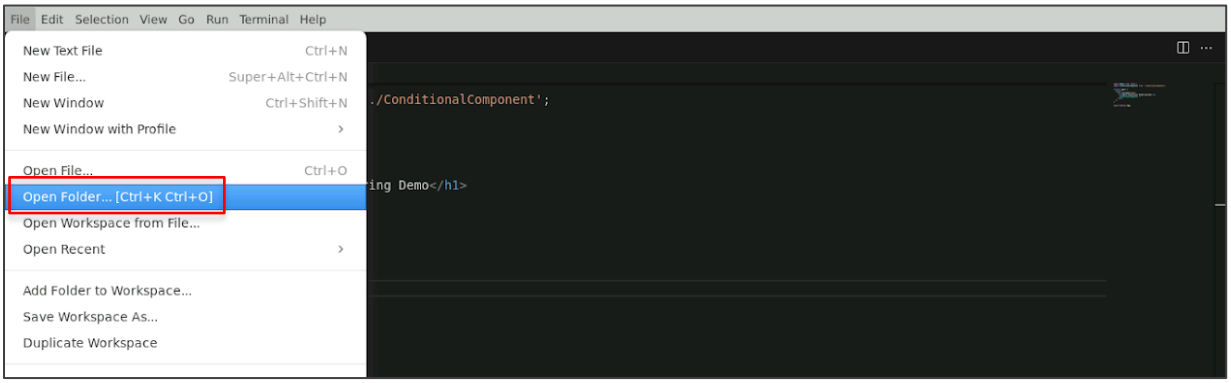
Prerequisites: None

Steps to be followed:

1. Write a JavaScript program using promises
2. Execute the program and verify the functionality of promises

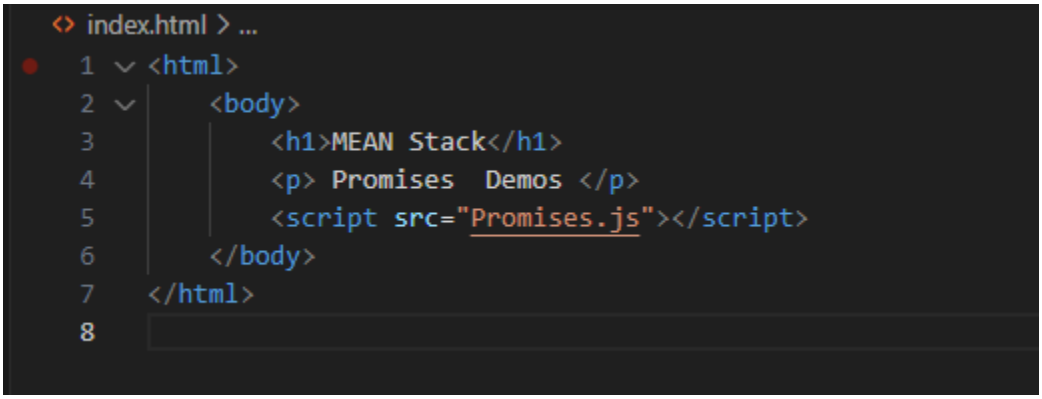
Step 1: Write a JavaScript program using promises

1.1 Navigate to Visual Studio Code, right-click on the **File** menu of the code editor, and select the **Open Folder** option:



1.2 Write the given code in **index.html**

```
<html>
  <body>
    <h1>MEAN Stack</h1>
    <p> Promises  Demos </p>
    <script src="Promises.js"></script>
  </body>
</html>
```



1.3 Click on the **src** folder of the project, select the **New File** option, enter the filename as **Promises.js**, and write the code shown below:

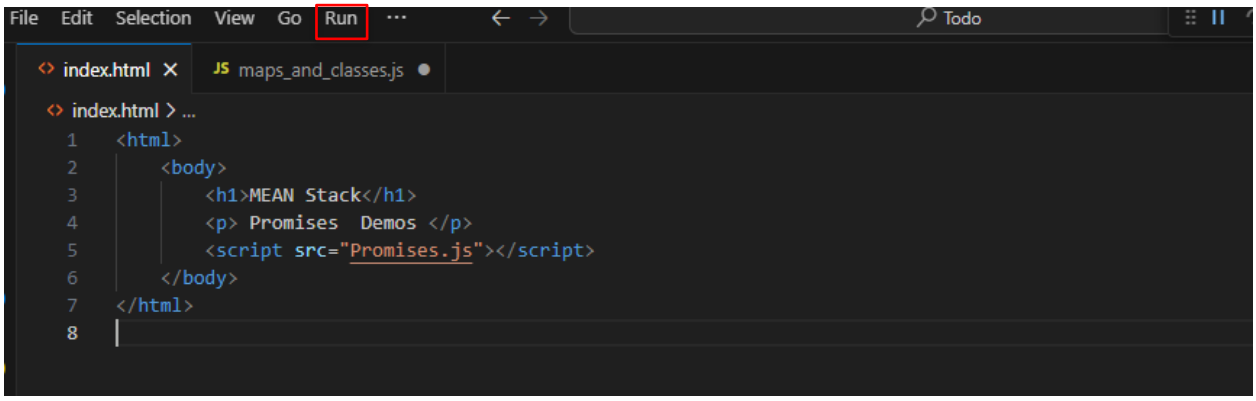
```
function flippingACoin() {
 return new Promise((resolve, reject) => {
  const val = Math.round(Math.random() * 1); // 0 or 1, at random
  val ? resolve('Heads!!') : reject('Tails!!');
 });
}
async function result() {
 try {
  const result = await flippingACoin();
  console.log(result);
 } catch(err) {
  console.log(err);
 }
}
result();
```

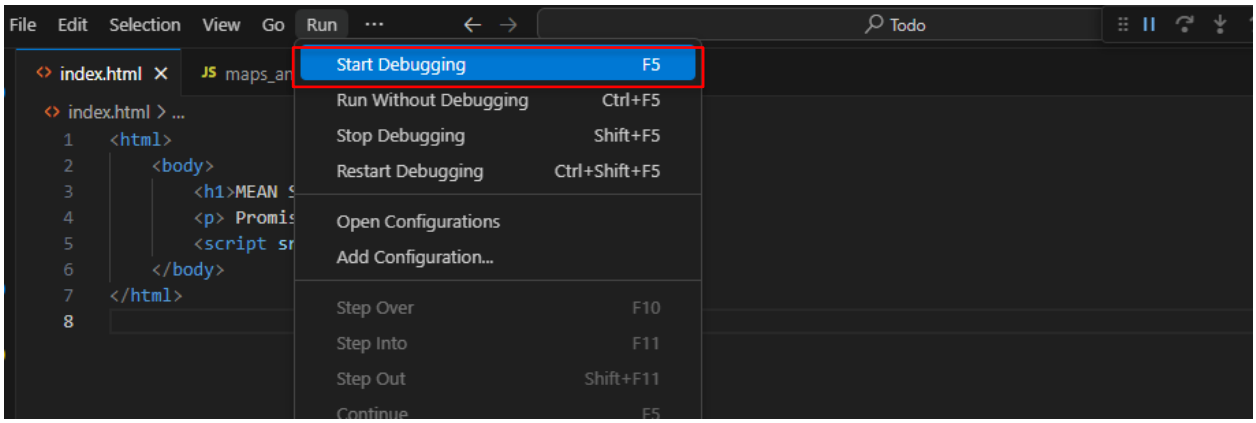
result();
result();
result();
result();

```
JS maps_and_classes.js > ...
1   function flippingACoin() {
2     return new Promise((resolve, reject) => {
3       const val = Math.round(Math.random() * 1); // 0 or 1, at random
4
5       val ? resolve('Heads!!') : reject('Tails!!');
6     });
7   }
8
9   async function result() {
10    try {
11      const result = await flippingACoin();
12      console.log(result);
13    } catch(err) {
14      console.log(err);
15    }
16  }
17
18  result();
19  result();
20  result();
21  result();
22  result();
23
```

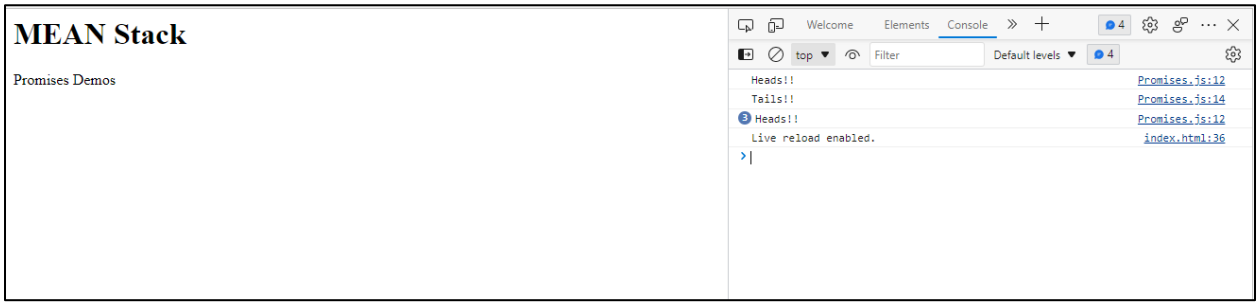
Step 2: Execute the program and verify the functionality of promises

2.1 Click on **Run** and then on **Start Debugging** to execute the JavaScript file





2.2 When the server starts running, right-click and select the **Inspect Element** option and click on the **Console** tab



By following the above steps, you have successfully implemented promises and asynchronous functions in JavaScript, demonstrating effective asynchronous control flow and error handling to enhance functionality and reliability within a web development framework.