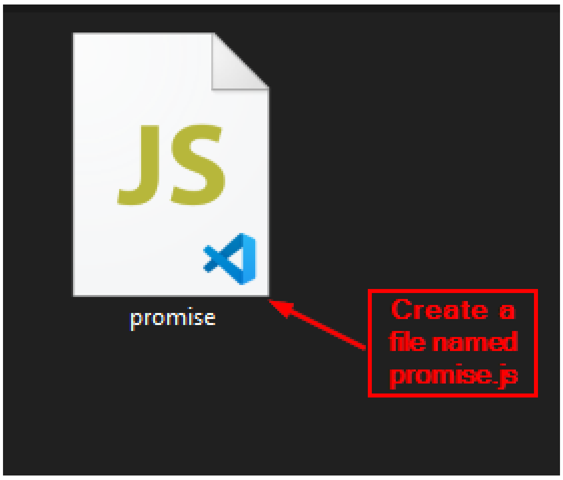
Module 2: Hands-On – 5

Performing Error Handling Using Promise

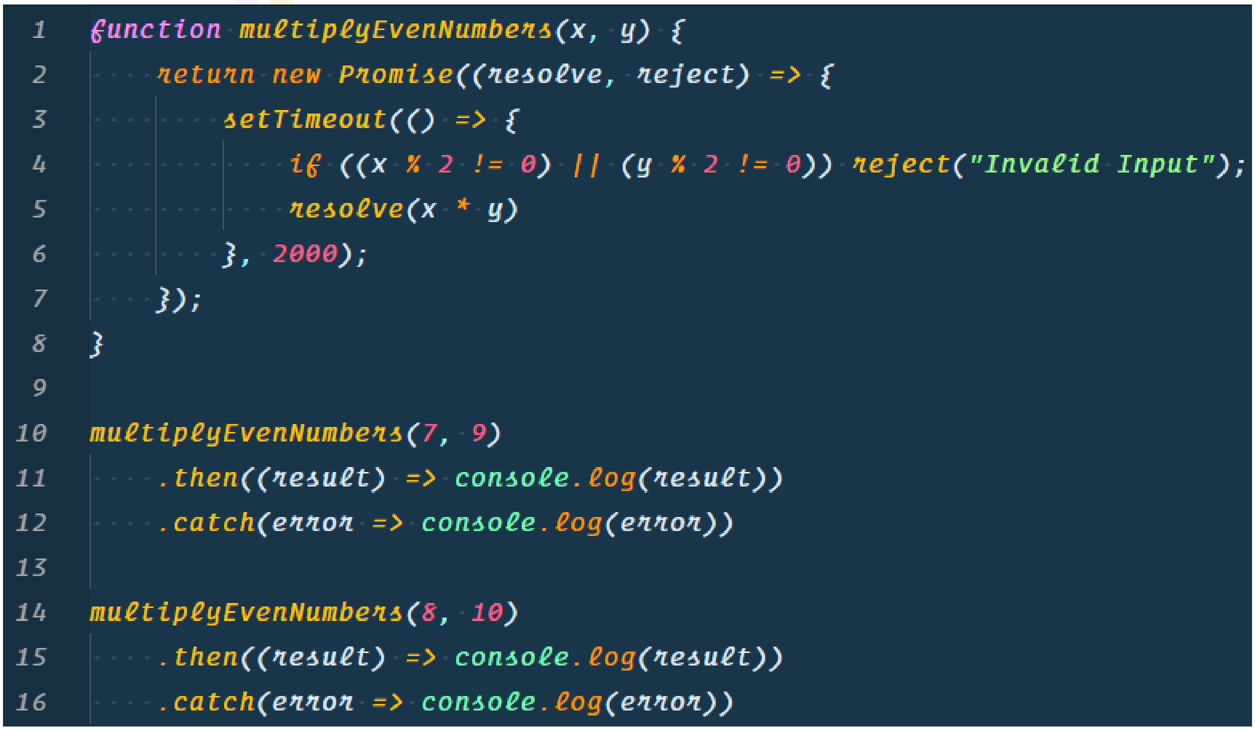
Step 1: Create a file named promise.js



Create a file named promise.js

Step 2: Open it in any text editor

Step 3: Type the following code:

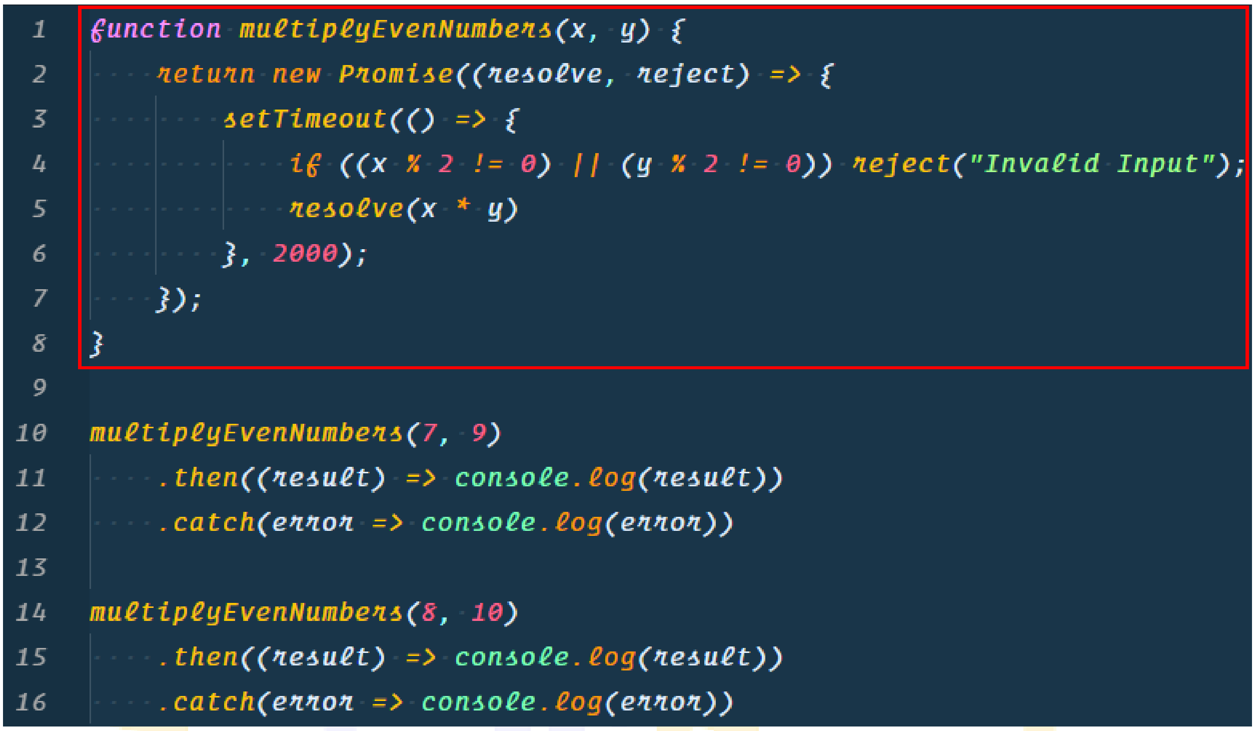


```javascript
function multiplyEvenNumbers(x, y) {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            if ((x % 2 != 0) || (y % 2 != 0)) reject("Invalid Input");
            resolve(x * y)
        }, 2000);
    });
}

multiplyEvenNumbers(7, 9)
    .then((result) => console.log(result))
    .catch(error => console.log(error))

multiplyEvenNumbers(8, 10)
    .then((result) => console.log(result))
    .catch(error => console.log(error))
```

Step 3.1: Create a function named **multiplyEvenNumbers** that takes two numbers as arguments: ***x*** and ***y*** return a new promise that waits for 2 seconds and then checks if either x or y is odd. If they are, then it calls the reject function with the error ‘Invalid Input’; else, it will call the resolve function with the result as x * y

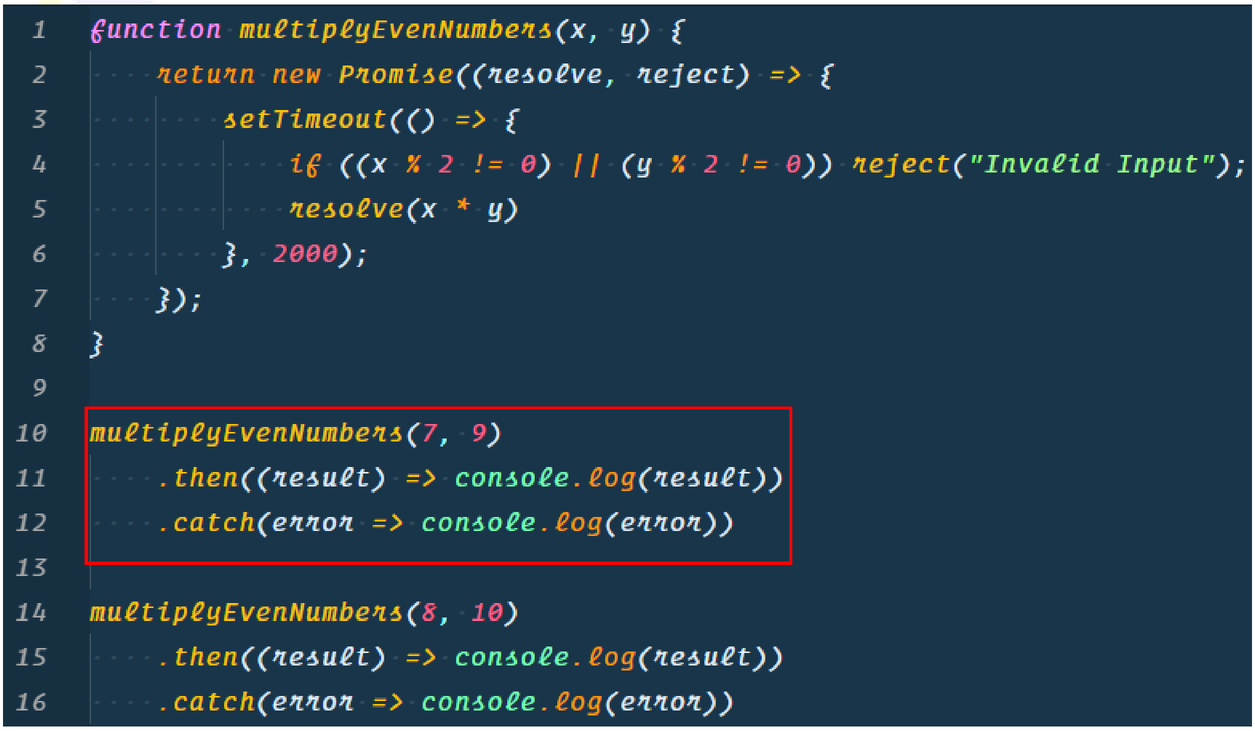


```
function multiplyEvenNumbers(x, y) {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            if ((x % 2 != 0) || (y % 2 != 0)) reject("Invalid Input");
            resolve(x * y)
        }, 2000);
    });
}

multiplyEvenNumbers(7, 9)
    .then((result) => console.log(result))
    .catch(error => console.log(error))

multiplyEvenNumbers(8, 10)
    .then((result) => console.log(result))
    .catch(error => console.log(error))
```

Step 3.2: Call the multiplyEvenNumbers function with a valid input (even numbers) and then call the .then method and pass a function that logs the result in the console and then call the .catch method pass a function that logs the error in the console.

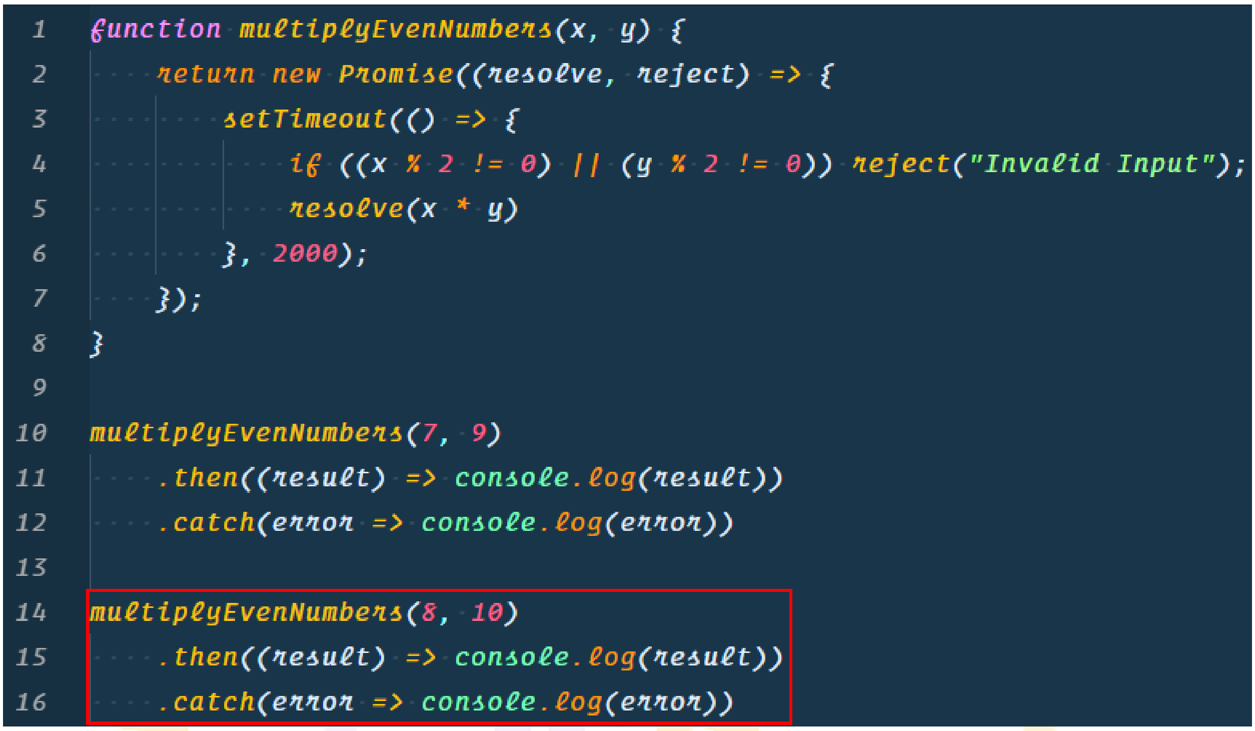


```
function multiplyEvenNumbers(x, y) {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            if ((x % 2 != 0) || (y % 2 != 0)) reject("Invalid Input");
            resolve(x * y)
        }, 2000);
    });
}

multiplyEvenNumbers(7, 9)
    .then((result) => console.log(result))
    .catch(error => console.log(error))

multiplyEvenNumbers(8, 10)
    .then((result) => console.log(result))
    .catch(error => console.log(error))
```

Step 3.3: Call the multiplyEvenNumbers function with an invalid input (odd numbers) and then call the .then method and pass a function that logs the result in the console; after that call the .catch method and pass a function that logs the error in the console

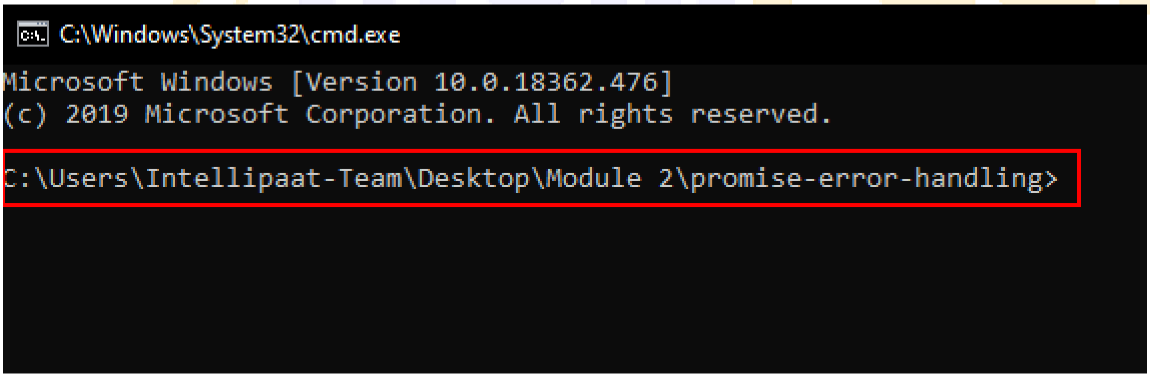


```
function multiplyEvenNumbers(x, y) {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            if ((x % 2 != 0) || (y % 2 != 0)) reject("Invalid Input");
            resolve(x * y)
        }, 2000);
    });
}

multiplyEvenNumbers(7, 9)
    .then((result) => console.log(result))
    .catch(error => console.log(error))

multiplyEvenNumbers(8, 10)
    .then((result) => console.log(result))
    .catch(error => console.log(error))
```

Step 4: Open the command prompt in the same directory as the file

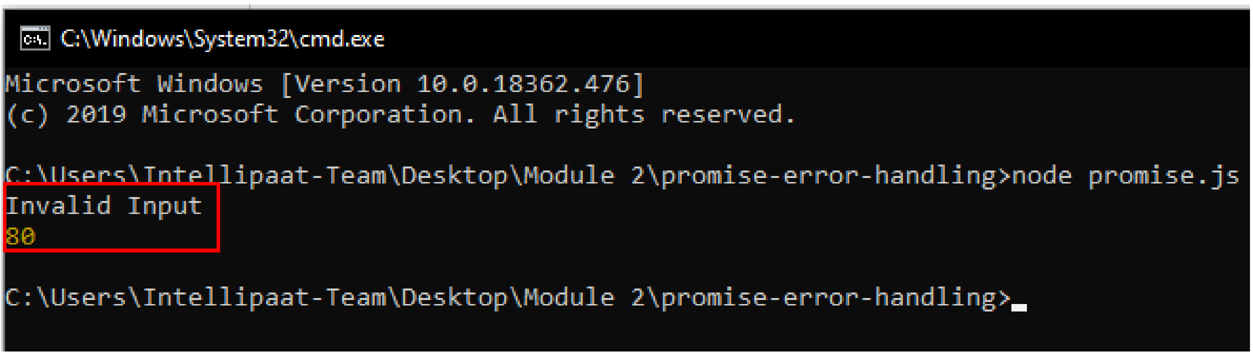


C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Intellipaat-Team\Desktop\Module 2\promise-error-handling>
```

Step 5: Run the file using the command ‘node promise.js’



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Intellipaat-Team\Desktop\Module 2\promise-error-handling>node promise.js
Invalid Input
80

C:\Users\Intellipaat-Team\Desktop\Module 2\promise-error-handling>
```