

Lesson 7 Demo 1

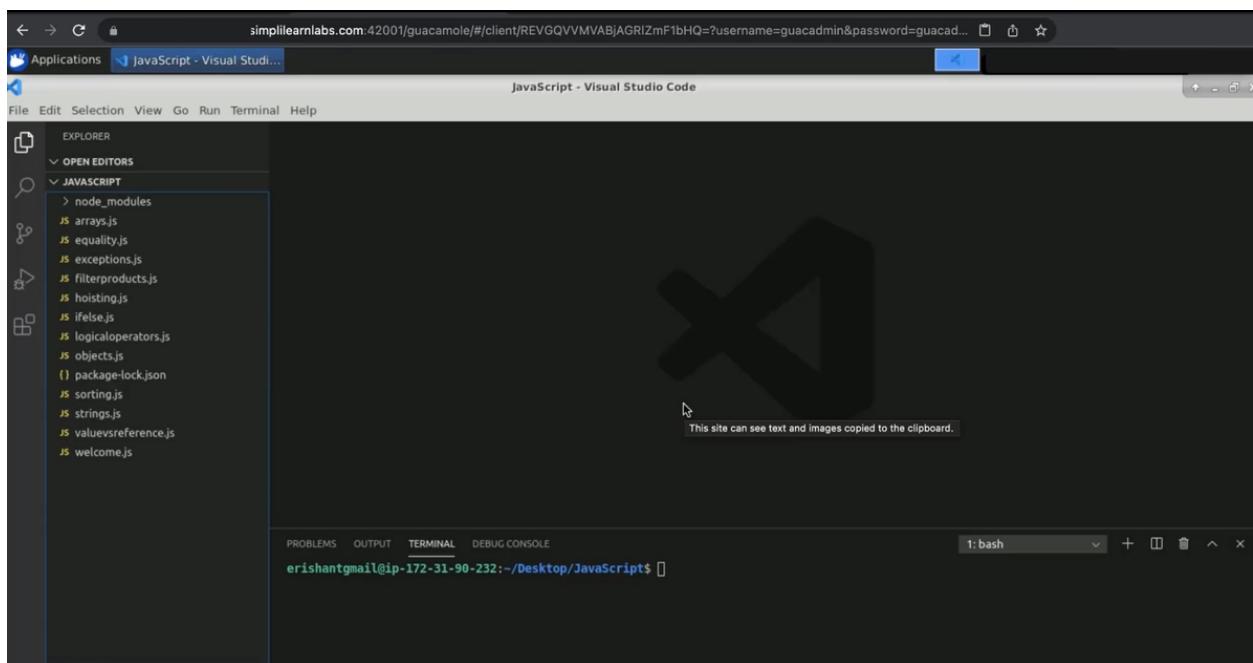
Async Programming in JS

Objective: To work with async programming in JS

Prerequisites: None

Steps to perform:

1. Open the visual studio code



2. Create a file name async.js

The screenshot shows a Visual Studio Code interface. The title bar indicates the URL is `simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacad...`. The editor tab shows `async.js - JavaScript - Vi...`. The left sidebar has sections for EXPLORER, OPEN EDITORS, and JAVASCRIPT. Under JAVASCRIPT, several files are listed: node_modules, arrays.js, async.js (which is currently selected), equality.js, exceptions.js, filterproducts.js, hoisting.js, ifelse.js, logicaloperators.js, objects.js, package-lock.json, sorting.js, strings.js, valuevsreference.js, and welcome.js. The main editor area shows the file `JS async.js` with the content `1 |`. Below the editor are tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab shows the command `erishant@gmailip-172-31-90-232:~/Desktop/JavaScript$`.

3. Create a function to upload a file. Enter 'let uploadFile = () => "File Uploaded Successfully";'.

The screenshot shows the same Visual Studio Code interface as the previous one, but the code in the editor has changed. The file `JS async.js` now contains the following code:
`1 let uploadFile = () => "File Uploaded Successfully";`

The status bar at the bottom of the editor shows '1 UNSAVED' next to the file name. The rest of the interface, including the sidebar and terminal, remains the same.

4. Create one more function.

```
Enter 'let updateUserProfile = () => {  
    let result = uploadFile();  
    console.log("Result is: "+result);  
}
```

```
JS async.js •  
JS async.js > ...  
1 let uploadFile = () => "File Uploaded Successfully";  
2  
3 let updateUserProfile = () => {  
4     let result = uploadFile();  
5     console.log("Result is: "+result);  
6 }  
7  
8 |
```

5. Execute updateUserProfile() method.

```
JS async.js •  
JS async.js > ...  
1 let uploadFile = () => "File Uploaded Successfully";  
2  
3 let updateUserProfile = () => {  
4     let result = uploadFile();  
5     console.log("Result is: "+result);  
6 }  
7  
8 updateUserProfile();
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript\$ node async.js

6. Enter 'node async.js' in terminal.

```

Edit Selection View Go Run Terminal Help
EXPLORER JS async.js X
OPEN EDITORS JS async.js > ...
JAVASCRIPT > node_modules
JS arrays.js JS async.js
JS equality.js JS exceptions.js
JS filterproducts.js JS hoisting.js
JS ifelse.js JS logicaloperators.js
JS objects.js () package-lock.json
JS sorting.js JS strings.js
JS valuesreference.js JS welcome.js
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
erishant@gmail.com@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
1: bash +
> OUTLINE

```

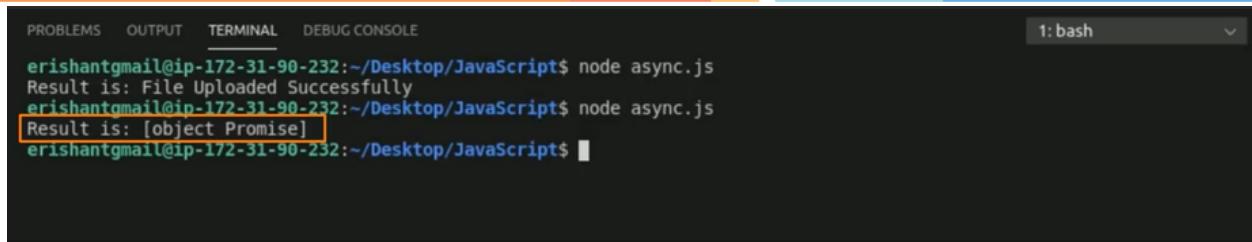
7. Enter 'async' keyword with uploadFile() function.

```

Edit Selection View Go Run Terminal Help
EXPLORER JS async.js X
OPEN EDITORS JS async.js > [e] uploadFile
JAVASCRIPT > node_modules
JS arrays.js JS async.js
JS equality.js JS exceptions.js
JS filterproducts.js JS hoisting.js
JS ifelse.js JS logicaloperators.js
JS objects.js () package-lock.json
JS sorting.js JS strings.js
JS valuesreference.js JS welcome.js

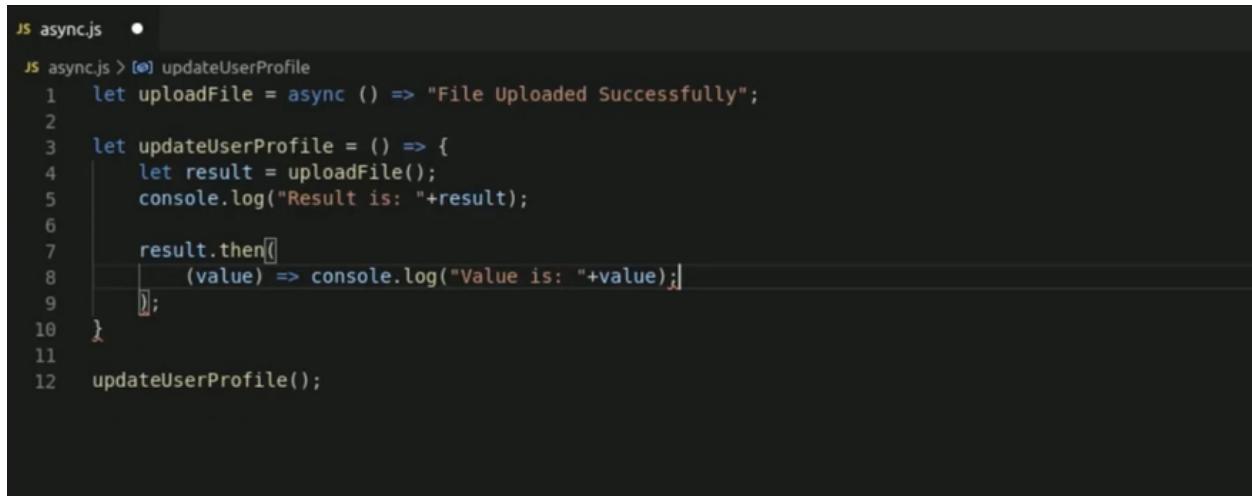
```

8. Run the code in terminal.



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

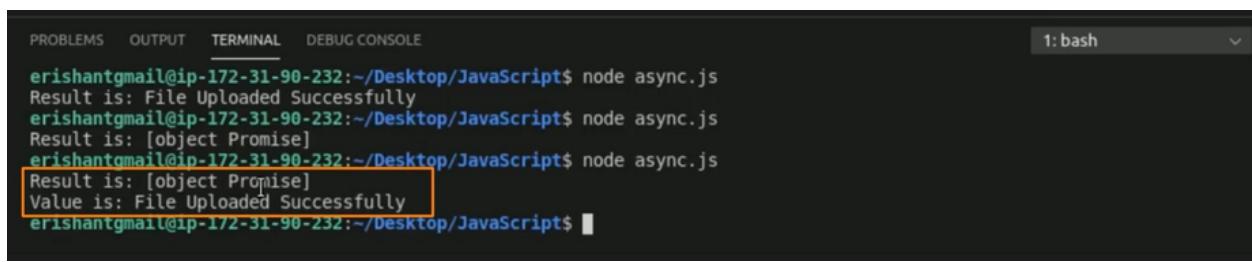
9. Add another method. Enter result.then(
(value) => console.log("Value is: "+value);
);



```
JS async.js ●

JS async.js > [●] updateUserProfile
1   let uploadFile = async () => "File Uploaded Successfully";
2
3   let updateUserProfile = () => {
4     let result = uploadFile();
5     console.log("Result is: "+result);
6
7     result.then([
8       (value) => console.log("Value is: "+value)];
9   ];
10 }
11
12 updateUserProfile();
```

10. Run the code in terminal.



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
Value is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

11. Enter 'async' keyword with updateUserProfile() function.

```
JS async.js ●  
JS async.js > [0] updateUserProfile  
1  let uploadFile = async () => "File Uploaded Successfully";  
2  
3  let updateUserProfile = async () => {  
4      let result = uploadFile();  
5      /*console.log("Result is: "+result);  
6  
7      result.then(  
8          (value) => console.log("Value is: "+value)  
9      );  
10 }  
11  
12  updateUserProfile();
```

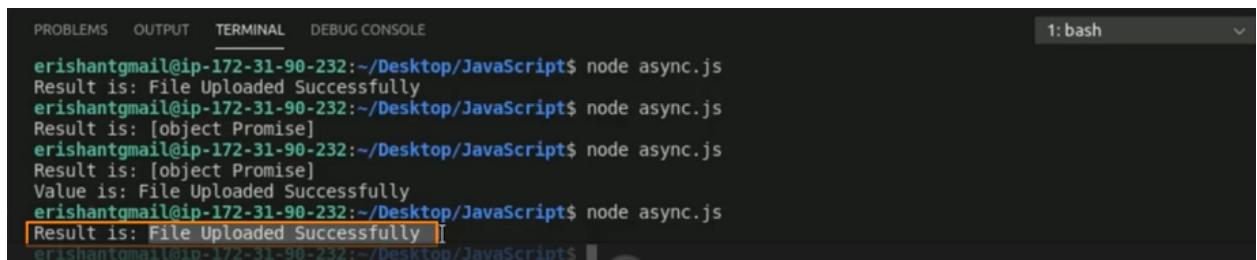
12. Enter 'await' in result upload file function().

```
JS async.js ●  
JS async.js > [0] updateUserProfile  
1  let uploadFile = async () => "File Uploaded Successfully";  
2  
3  let updateUserProfile = async () => {  
4      let result = await uploadFile();  
5      /*console.log("Result is: "+result);  
6  
7      result.then(  
8          (value) => console.log("Value is: "+value)  
9      );*/  
10 }  
11  
12  updateUserProfile();
```

13. Enter 'console.log("Result is: "+result);

```
JS async.js  X
JS async.js > [e] updateUserProfile
1   let uploadFile = async () => "File Uploaded Successfully";
2
3   let updateUserProfile = async () => [
4     let result = await uploadFile();
5     console.log("Result is: "+result);
6
7     /*console.log("Result is: "+result);
8
9     result.then(
10       (value) => console.log("Value is: "+value)
11     );*/
12   ]
13
14   updateUserProfile();
```

14. Run the code in terminal.



```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
Value is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
```

15. Create new promise object. Enter ' let myPromise = new Promise(
function(resolve, reject){ }).

```

JS async.js ●
JS async.js > [o] myPromise
1  let uploadFile = async () => "File Uploaded Successfully";
2
3  let updateUserProfile = async () => {
4    let result = await uploadFile();
5    console.log("Result is: "+result);
6
7    /*console.log("Result is: "+result);
8
9    result.then(
10      (value) => console.log("Value is: "+value)
11    );*/
12  }
13
14 updateUserProfile();
15
16 let myPromise = new Promise(
17   function(resolve, reject){
18     // Algo goes here
19
20   }
21 );

```

16. Inside {}, Enter '//Algo goes here

```

let isDataFetched =true;
if (isDataFetched){
  resolve();
}else{
  reject();
}.

```

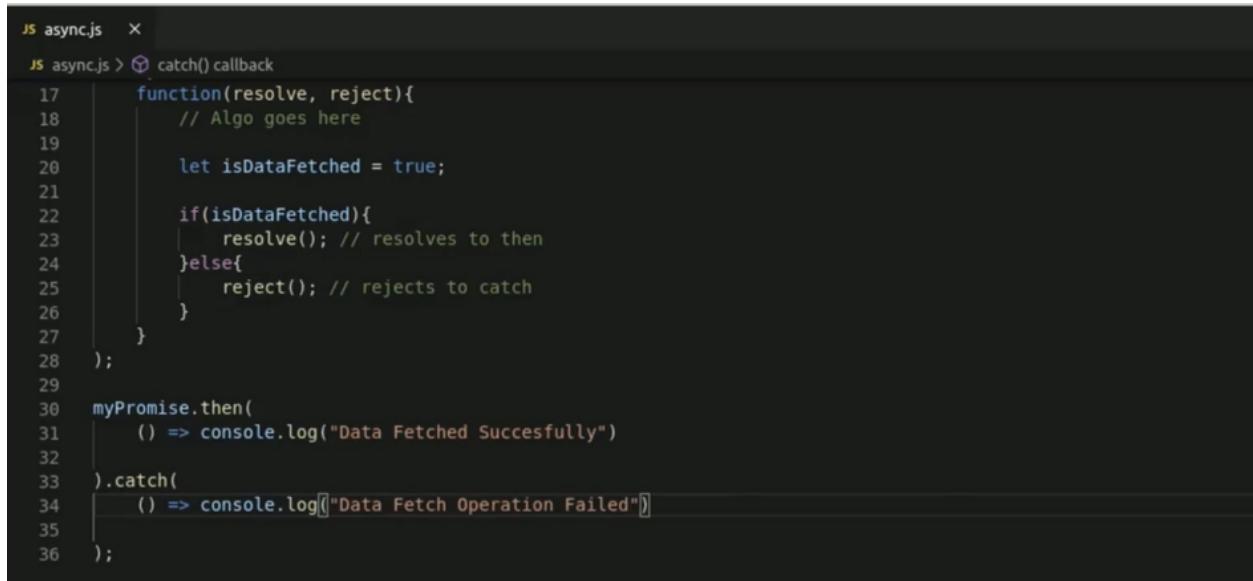
```

JS async.js ●
JS async.js > [o] myPromise > ⓘ <function>
11   );*/
12 }
13
14 updateUserProfile();
15
16 let myPromise = new Promise(
17   function(resolve, reject){
18     // Algo goes here
19
20     let isDataFetched = true;
21
22     if(isDataFetched){
23       resolve();
24     }else{
25       reject();
26     }
27   }
28 );

```

17. Enter 'myPromise.then(

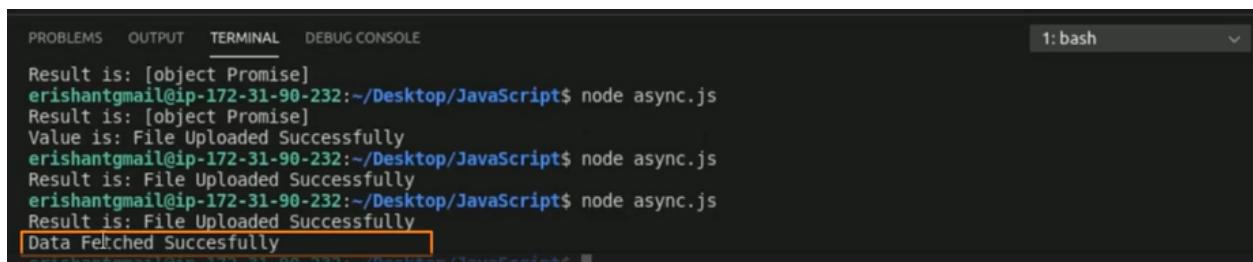
```
() => console.log("Data Fetched Successfully")
).catch(
() => console.log("Data Fetched Operation Failed")
);.
```



The screenshot shows a code editor with a dark theme. A code completion suggestion is displayed over the code. The suggestion is for a 'catch()' callback, which is part of a promise chain. The suggestion text is: 'JS async.js > catch() callback'. The code being completed is:

```
JS async.js > catch() callback
    function(resolve, reject){
        // Algo goes here
        let isDataFetched = true;
        if(isDataFetched){
            resolve(); // resolves to then
        }else{
            reject(); // rejects to catch
        }
    );
}
myPromise.then(
    () => console.log("Data Fetched Succesfully")
).catch(
    () => console.log(["Data Fetch Operation Failed"])
);
```

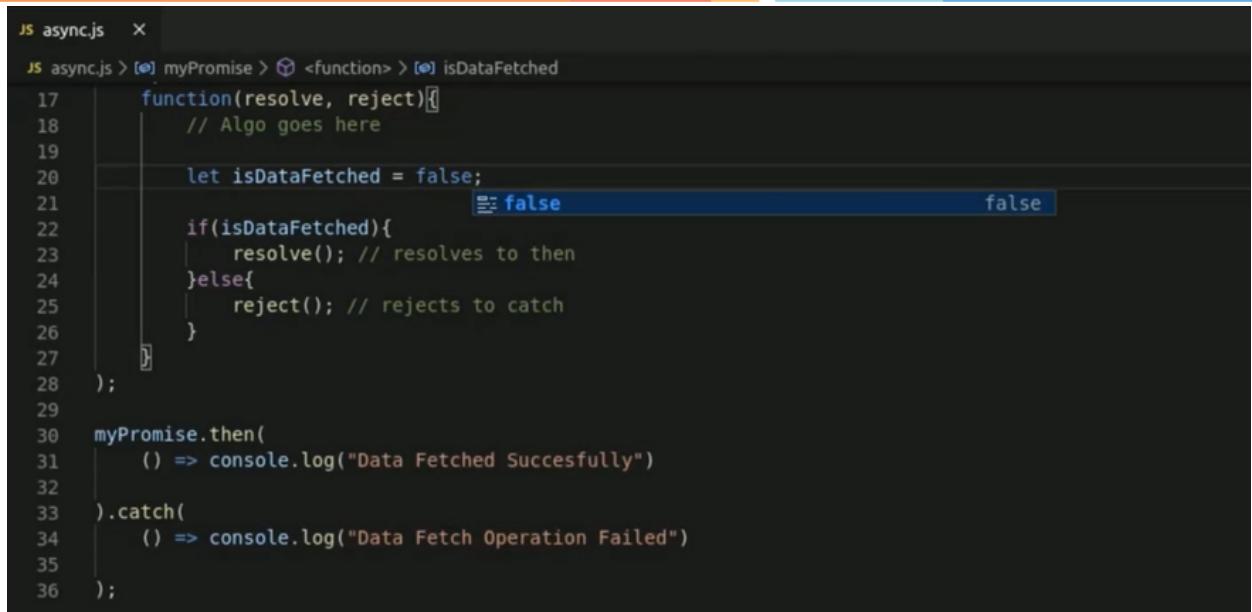
18. Run the code in terminal.



The screenshot shows a terminal window with the following output:

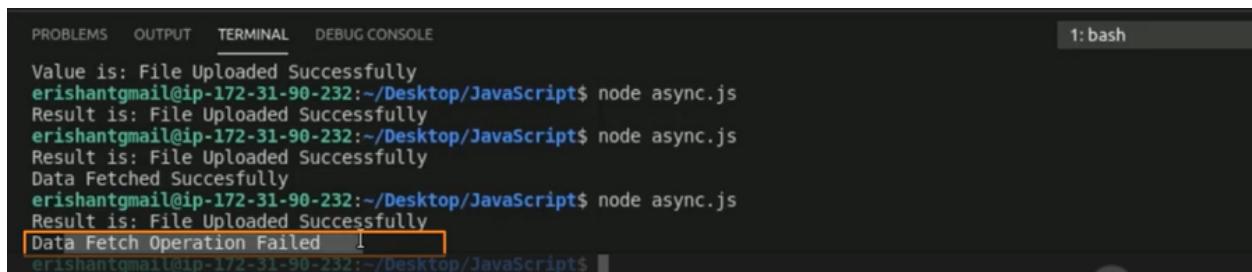
```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Result is: [object Promise]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
Value is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
Data Fetched Succesfully
```

19. Enter 'isDataFetched = False;':



```
JS async.js  X
JS async.js > [0] myPromise > ↴ <function> > [0] isDataFetched
17   function(resolve, reject){
18     // Algo goes here
19
20     let isDataFetched = false;
21     ↳ false
22     if(isDataFetched){
23       resolve(); // resolves to then
24     }else{
25       reject(); // rejects to catch
26     }
27   ]
28 );
29
30 myPromise.then(
31   () => console.log("Data Fetched Successfully")
32 )
33 .catch(
34   () => console.log("Data Fetch Operation Failed")
35
36 );
```

20. Run the code in terminal.



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: bash
Value is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
Data Fetched Successfully
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
Data Fetch Operation Failed ↳
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$
```