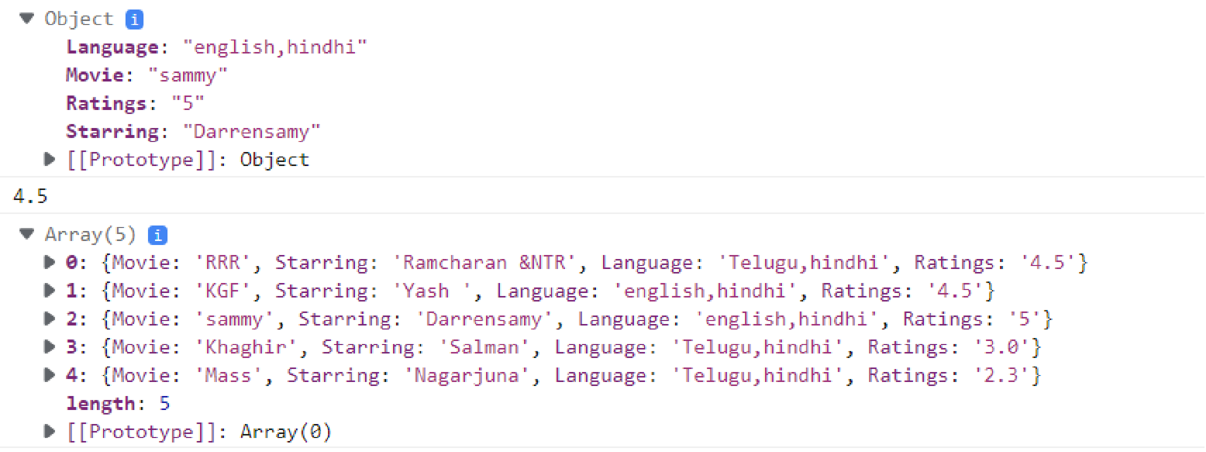
5.a Course Name: Javascript Module Name: Creating Arrays, Destructuring Arrays, Accessing Arrays, Array Methods Create an array of objects having movie details. The object should include the movie name, starring, language, and ratings. Render the details of movies on the page using the array.

Program:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Java Script</title>
</head>
<body>
<script type="text/javascript">
const mobiles=[{ "Movie":"RRR","Starring":"Ramcharan &
NTR","Language":"Telugu,hindhi","Ratings":"4.5"},
{ "Movie":"KGF","Starring":"Yash ","Language":"english,hindhi","Ratings":"4.5"},
{"Movie":"sammy","Starring":"Darrensamy","Language":"english,hindhi","Ratings":"5},
{"Movie":"Khaghir","Starring":"Salman","Language":"Telugu,hindhi","Ratings":"3.0"},  {
"Movie":"Mass","Starring":"Nagarjuna","Language":"Telugu,hindhi","Ratings":"2.3"},  ];
console.log(mobiles[2]);
console.log(mobiles[0].Ratings);
console.log(mobiles);
document.write(mobiles);
</script>
</body>
</html>
```

OUTPUT:



```
▼ Object ℹ
    Language: "english,hindhi"
    Movie: "sammy"
    Ratings: "5"
    Starring: "Darrensamy"
  ▶ [[Prototype]]: Object
 4.5
▼ Array(5) ℹ
  ▶ 0: {Movie: 'RRR', Starring: 'Ramcharan &NTR', Language: 'Telugu,hindhi', Ratings: '4.5'}
  ▶ 1: {Movie: 'KGF', Starring: 'Yash ', Language: 'english,hindhi', Ratings: '4.5'}
  ▶ 2: {Movie: 'sammy', Starring: 'Darrensamy', Language: 'english,hindhi', Ratings: '5'}
  ▶ 3: {Movie: 'Khaghir', Starring: 'Salman', Language: 'Telugu,hindhi', Ratings: '3.0'}
  ▶ 4: {Movie: 'Mass', Starring: 'Nagarjuna', Language: 'Telugu,hindhi', Ratings: '2.3'}
    length: 5
  ▶ [[Prototype]]: Array(0)
```

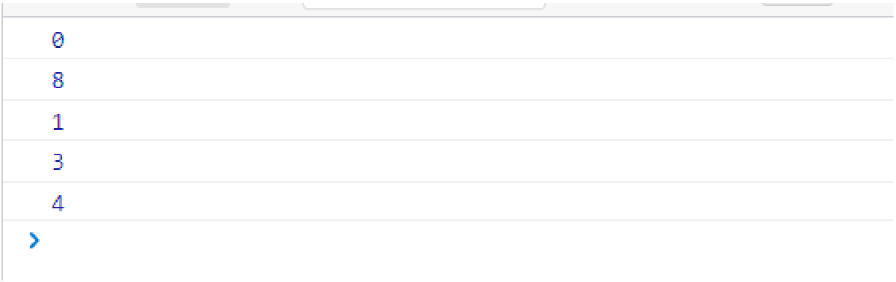
5.b Course Name: Javascript

Module Name: Introduction to Asynchronous Programming, Callbacks, Promises, Async and Await, Executing Network Requests using Fetch API Simulate a periodic stock price change and display on the console. Hints: (i) Create a method which returns a random number - use Math.random, floor and other methods to return a rounded value. (ii) Invoke the method for every three seconds and stop when the count is 5 – use the setInterval method. (iii) Since setInterval is an async method, enclose the code in a Promise and handle the response generated in a success callback. (iv) The random value returned from the method every time can be used as a stock price and displayed on the console.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-
scale=1.0">   <title>Exp__5b</title>
</head>
<body>
 <script>
let c=0;
while(c<5)
{   var sai = new Promise(function (resolve, reject) { setTimeout(function () {
 var a=Math.floor(Math.random() * 10); resolve(a);  }, 3000);  });
 sai.then( function(data) {
console.log(data);
}, function (error)  {
console.log(error);
 } );
c+=1;
}
 </script>
</body>
</html>
```

OUTPUT:



```
0
8
1
3
4
>
```

6.a Course Name: Node.js

Module Name: How to use Node.js and Verify how to execute different functions successfully in the Node.js platform

AIM: to use Node.js Verify how to execute different functions successfully in the Node.js platform

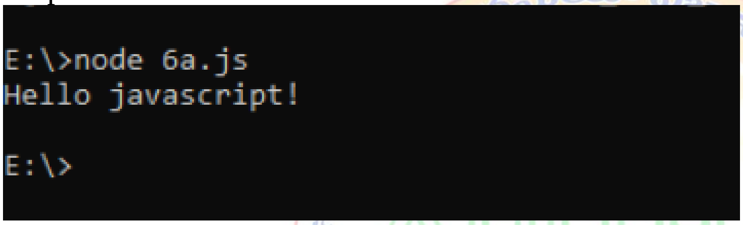
How to use node.js

Step 1: Create a folder NodeJS in D drive and create a new JavaScript file, **first.js** inside the folder. Type the code inside the JavaScript file.
Step 2: Navigate to the created NodeJS folder in the NodeJS command prompt and execute the JavaScript file, first.js using the **node** command.
Step 3: After the successful interpretation of the code, we can see the output in the Node.js command prompt .

Program:

```
function tester() {
  console.log("Hello javascript!");
}
tester();
```

Output:



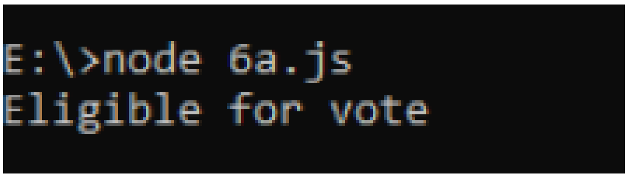
```
E:\>node 6a.js
Hello javascript!

E:\>
```

Modified program:

```
function tester() {
  var age=25;
  if(age>=18) {
    msg = "Eligible for vote";
  } else {
    msg= "not eligible for vote";  }
  console.log(msg);
}
tester();
```

output:



```
E:\>node 6a.js
Eligible for vote
```

6.b Course Name: Node.js Module Name: Create a web server in Node.js Write a program to show the workflow of JavaScript code executable by creating web server in Node.js

AIM: To Create a web server in Node.js Write a program to show the workflow of JavaScript code executable by creating web server in Node.js

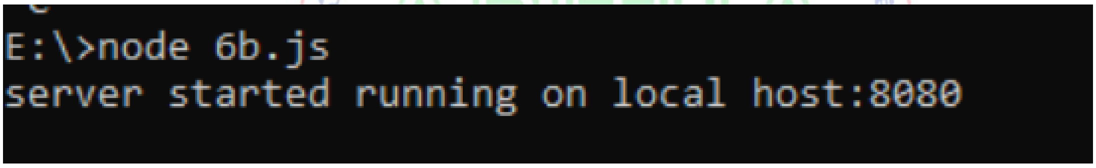
Steps to create web server:

Step 1: Create a new JavaScript file and include the HTTP module.
Step 2: Use the createServer() method of the HTTP module to create a web server.
Step 3: Save the file and start the server using the **node** command. When the file executes successfully, we can observe the following output in the console.
Step 4: We will observe the following in the browser.

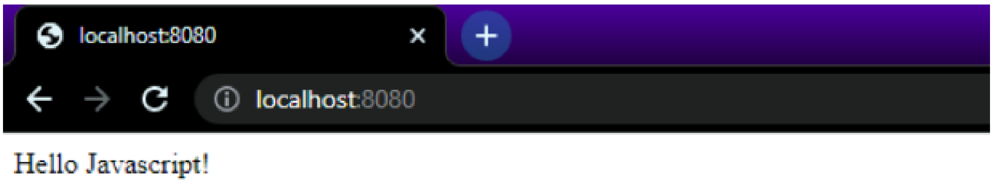
Program:

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.end('Hello Javascript!');
}).listen(8080);
console.log("server started running on local host:8080 ");
```

Output:



```
E:\>node 6b.js
server started running on local host:8080
```



Hello Javascript!

6.c Course Name: Node.js

Module Name: Modular programming in Node.js

Write a Node.js module to show the workflow of Modularization of Node application

AIM: To write a Node.js module to show the workflow of Modularization of Node application

Program:

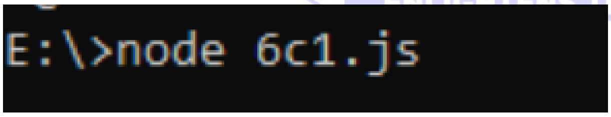
6c.js:

```
exports.myDateTime = function () {
  return Date();
};
```

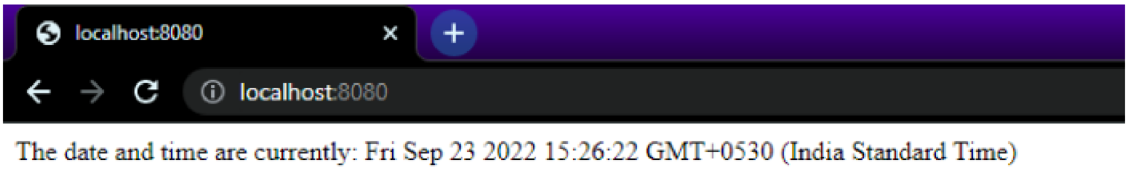
6c1.js:

```
var http = require('http');
var dt = require('./6c');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("The date and time are currently: " + dt.myDateTime());
  res.end();
}).listen(8080);
```

Output:



```
E:\>node 6c1.js
```



The date and time are currently: Fri Sep 23 2022 15:26:22 GMT+0530 (India Standard Time)