Lesson 05 Demo 05

Implementing AJAX Calls

Objective: To implement AJAX calls

Tools required: Ubuntu Linux VM

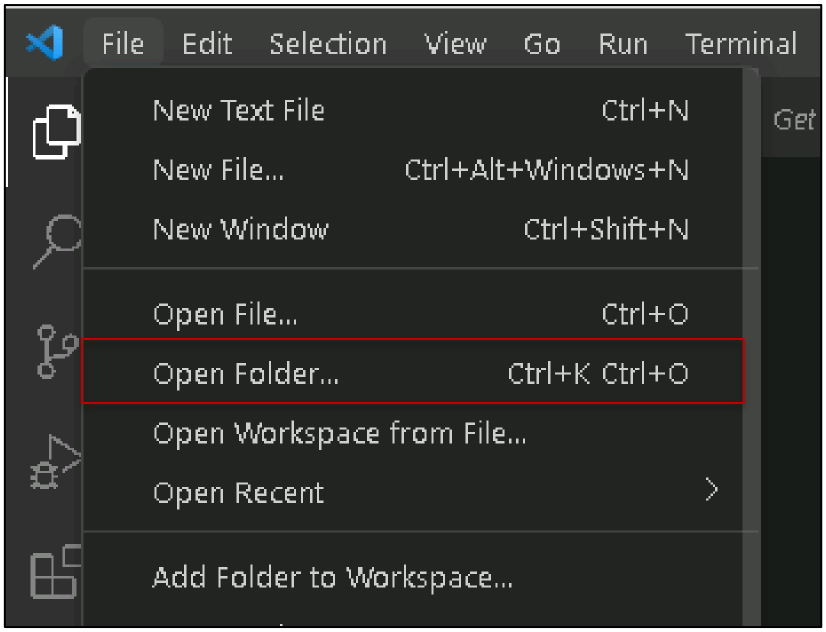
Prerequisites: Visual Studio Code, Browser

Steps to be followed:

1. Writing code for AJAX
2. Executing and verifying the working of AJAX calls

Step 1: Writing a code for AJAX, fetch and promise:

1.1 Open Visual Studio Code and Right click on the **File** menu of the code editor and select ***Open Folder** option*:

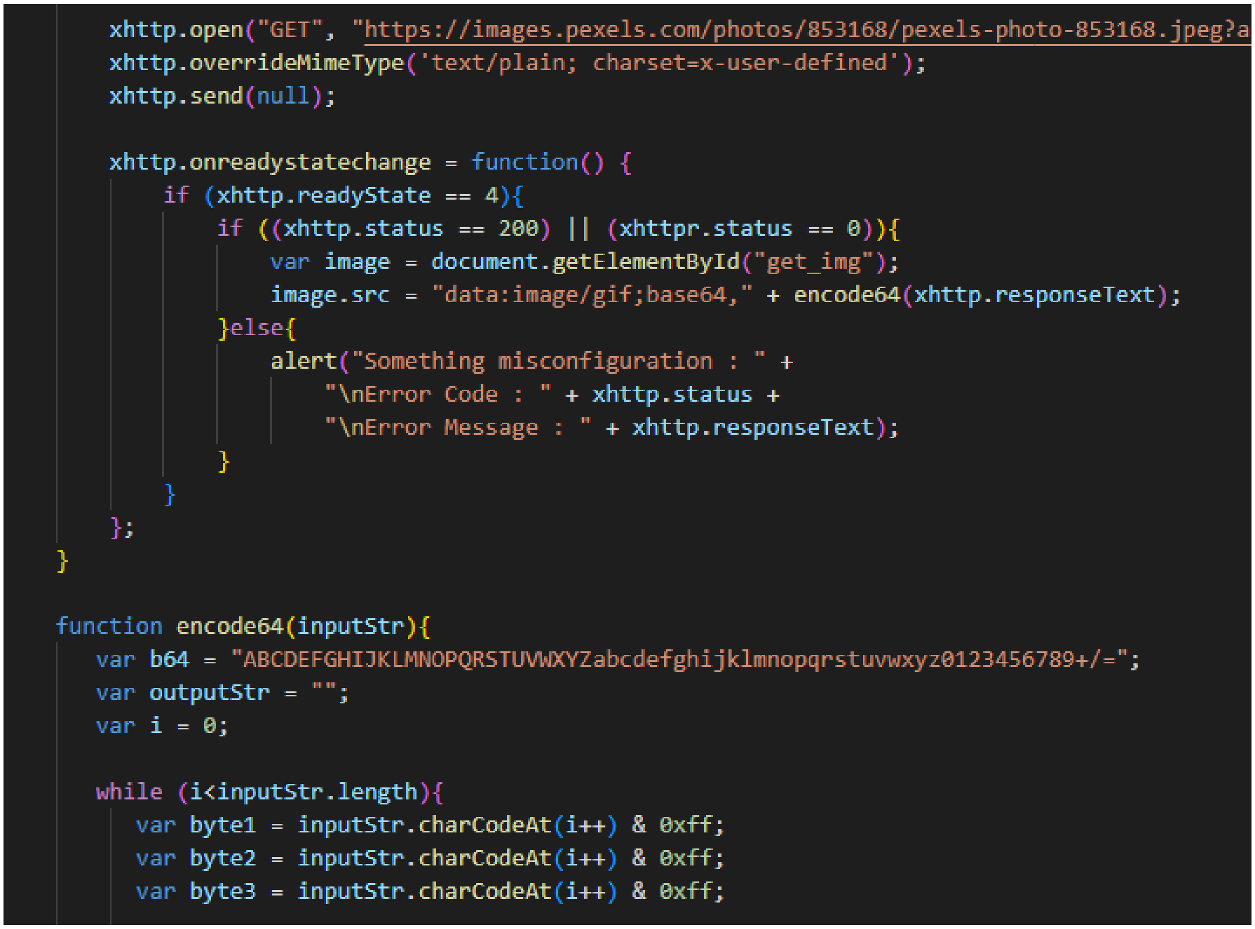

File
Edit
Selection
View
Go
Run
Terminal
New Text File Ctrl+N
New File... Ctrl+Alt+Windows+N
New Window Ctrl+Shift+N
Open File... Ctrl+O
Open Folder... Ctrl+K Ctrl+O
Open Workspace from File...
Open Recent
Add Folder to Workspace...
Get

1.2 Right click on the **src** folder of the project and select ***New File*** and enter the filename as **ajax_demo1.html**

1.3 Write the code shown below in the **ajax_demo1.html**:

```
<html>
<body>
    <header>
        <h1>MEAN Stack</h1>
        <p> Ajax without Fetch and Promise</p>
        </header>

<script language="javascript" type="text/javascript">
  function ajax_call_demo( ){
    if (window.XMLHttpRequest ) {
      {
        xhttp = new XMLHttpRequest();
      }
    } else {
      alert("Your browser does not support XMLHttpRequest...!");
    }

    xhttp.open("GET", "https://images.pexels.com/photos/853168/pexels-photo-
853168.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500", true);  // Make sure file is in same
server
    xhttp.overrideMimeType('text/plain; charset=x-user-defined');
    xhttp.send(null);

    xhttp.onreadystatechange = function() {
      if (xhttp.readyState == 4){
        if ((xhttp.status == 200) || (xhttpr.status == 0)){
          var image = document.getElementById("get_img");
          image.src = "data:image/gif;base64," + encode64(xhttp.responseText);
        }else{
          alert("Something misconfiguration : " +
            "\nError Code : " + xhttp.status +
            "\nError Message : " + xhttp.responseText);
          }
       }
     };
  }

  function encode64(inputStr){
    var b64 =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";
    var outputStr = "";
    var i = 0;

    while (i<inputStr.length){
      var byte1 = inputStr.charCodeAt(i++) & 0xff;
      var byte2 = inputStr.charCodeAt(i++) & 0xff;
      var byte3 = inputStr.charCodeAt(i++) & 0xff;

      var enc1 = byte1 >> 2;
      var enc2 = ((byte1 & 3) << 4) | (byte2 >> 4);

      var enc3, enc4;
      if (isNaN(byte2)){
       enc3 = enc4 = 64;
      } else{
       enc3 = ((byte2 & 15) << 2) | (byte3 >> 6);
       if (isNaN(byte3)){
         enc4 = 64;
       } else {
          enc4 = byte3 & 63;
       }
      }
      outputStr +=  b64.charAt(enc1) + b64.charAt(enc2) + b64.charAt(enc3) +
b64.charAt(enc4);
    }
    return outputStr;
  }
</script>

<button onClick="ajax_call_demo()">Click here to get an image</button><br />
```

<img id="get_img" />
</body>
</html>

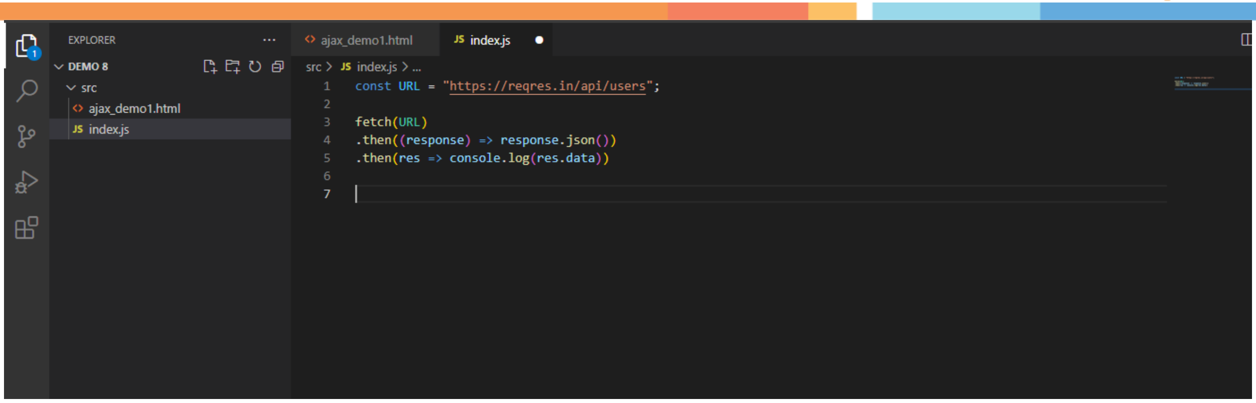


```
    xhttp.open("GET", "https://images.pexels.com/photos/853168/pexels-photo-853168.jpeg?a
    xhttp.overrideMimeType('text/plain; charset=x-user-defined');
    xhttp.send(null);

    xhttp.onreadystatechange = function() {
        if (xhttp.readyState == 4){
            if ((xhttp.status == 200) || (xhttpr.status == 0)){
                var image = document.getElementById("get_img");
                image.src = "data:image/gif;base64," + encode64(xhttp.responseText);
            }else{
                alert("Something misconfiguration : " +
                    "\nError Code : " + xhttp.status +
                    "\nError Message : " + xhttp.responseText);
            }
        }
    };
}

function encode64(inputStr){
   var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";
   var outputStr = "";
   var i = 0;

   while (i<inputStr.length){
      var byte1 = inputStr.charCodeAt(i++) & 0xff;
      var byte2 = inputStr.charCodeAt(i++) & 0xff;
      var byte3 = inputStr.charCodeAt(i++) & 0xff;
```

1.5 Right click on the **src** folder of the project and select ***New File*** option and enter the filename as **index.js.**

1.6 Write the code shown below in the **index.js**:

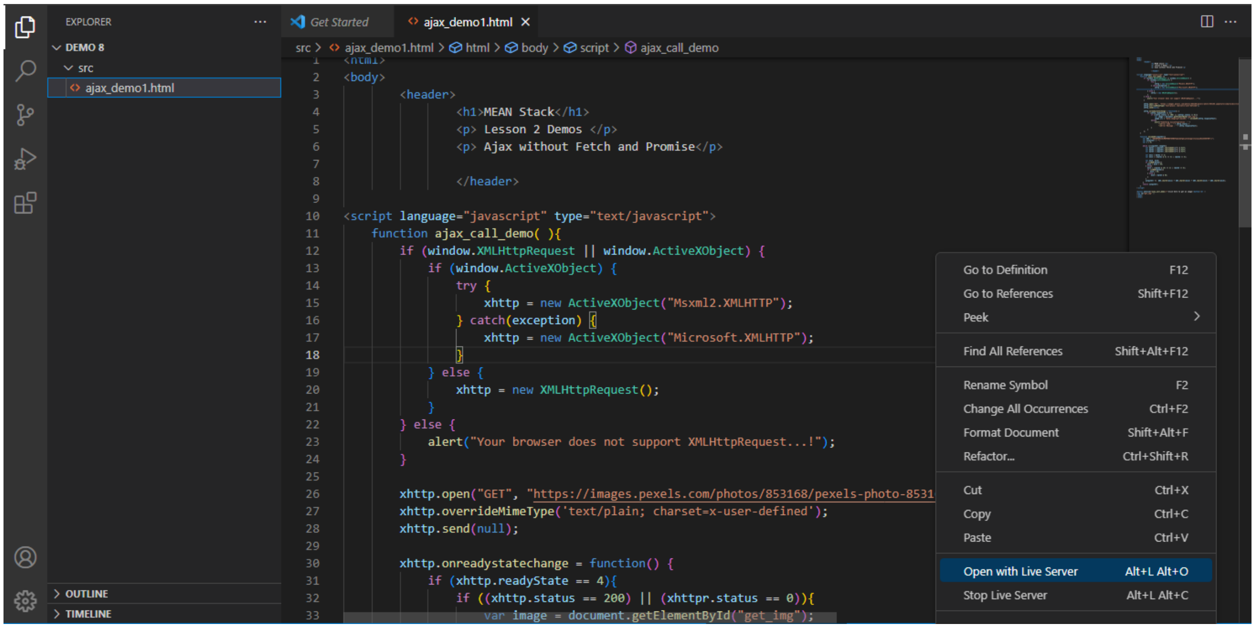
const URL = "https://reqres.in/api/users";
fetch(URL)
.then((response) => response.json())
.then(res => console.log(res.data))



```
const URL = "https://reqres.in/api/users";

fetch(URL)
.then((response) => response.json())
.then(res => console.log(res.data))

```

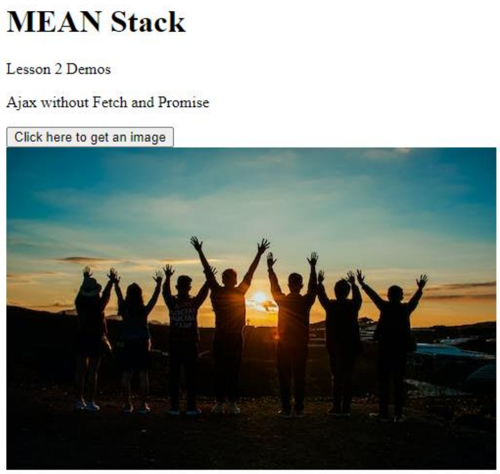
Step 2: Executing and verifying the working of AJAX calls, fetch, and promise:

2.1 Right click on the **ajax_demo1.html** file of the project and select ***Open with Live Server***


EXPLORER
DEMO 8
src
ajax_demo1.html
Get Started
ajax_demo1.html
src > ajax_demo1.html > html > body > script > ajax_call_demo
Go to Definition F12
Go to References Shift+F12
Peek
Find All References Shift+Alt+F12
Rename Symbol F2
Change All Occurrences Ctrl+F2
Format Document Shift+Alt+F
Refactor... Ctrl+Shift+R
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Open with Live Server Alt+L Alt+O
Stop Live Server Alt+L Alt+C
OUTLINE
TIMELINE

```
<html>
<body>
        <header>
                <h1>MEAN Stack</h1>
                <p> Lesson 2 Demos </p>
                <p> Ajax without Fetch and Promise</p>

                </header>

<script language="javascript" type="text/javascript">
    function ajax_call_demo( ){
        if (window.XMLHttpRequest || window.ActiveXObject) {
            if (window.ActiveXObject) {
                try {
                    xhttp = new ActiveXObject("Msxml2.XMLHTTP");
                } catch(exception) {
                    xhttp = new ActiveXObject("Microsoft.XMLHTTP");
                }
            } else {
                xhttp = new XMLHttpRequest();
            }
        } else {
            alert("Your browser does not support XMLHttpRequest...!");
        }

        xhttp.open("GET", "https://images.pexels.com/photos/853168/pexels-photo-8531
        xhttp.overrideMimeType('text/plain; charset=x-user-defined');
        xhttp.send(null);

        xhttp.onreadystatechange = function() {
            if (xhttp.readyState == 4){
                if ((xhttp.status == 200) || (xhttpr.status == 0)){
                    var image = document.getElementById("get_img");
```

2.2 Right click when the server starts running.



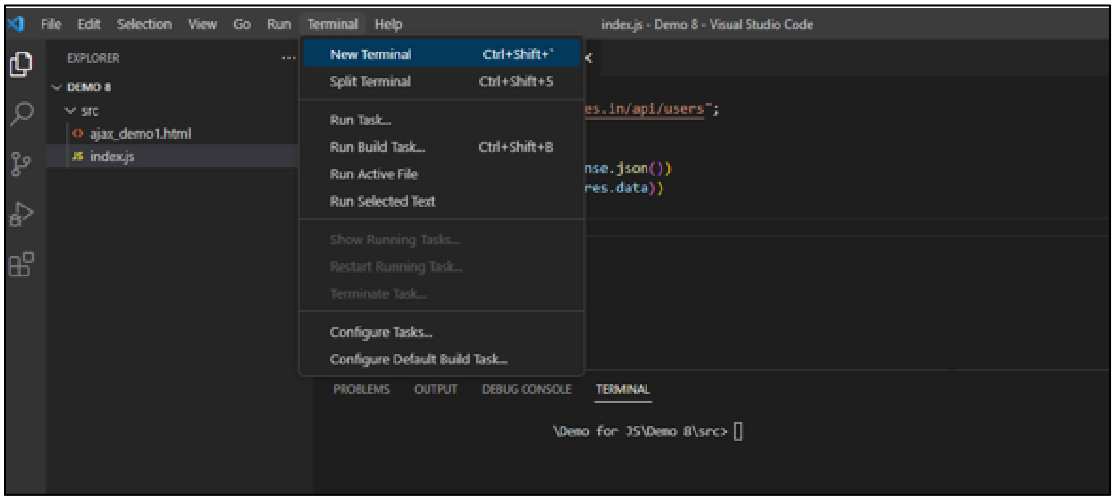
MEAN Stack

Lesson 2 Demos

Ajax without Fetch and Promise

Click here to get an image

2.3 Click on **Terminal**, and select the **New Terminal** option:

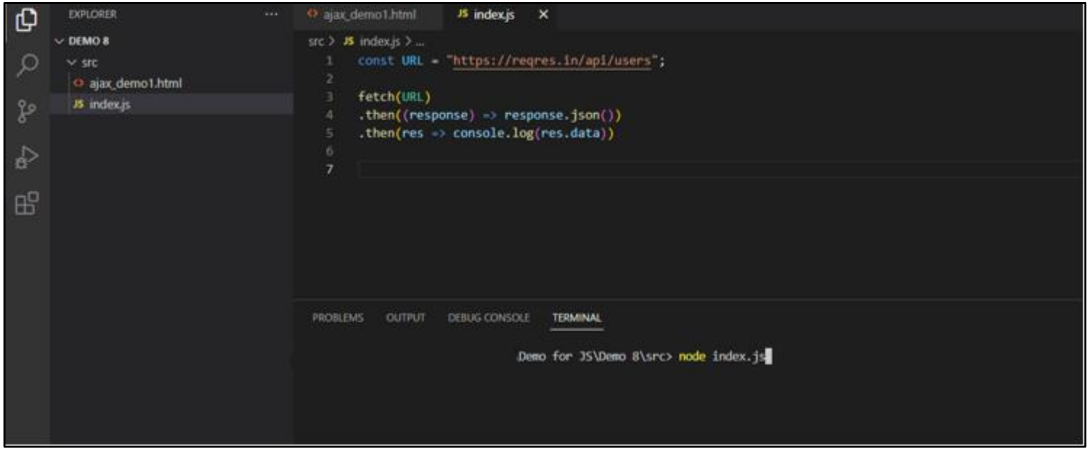


```
es.in/api/users";

nse.json())
res.data))
```

```
\Demo for JS\Demo 8\src>
```

2.4 Write the following command to execute **index.js** file

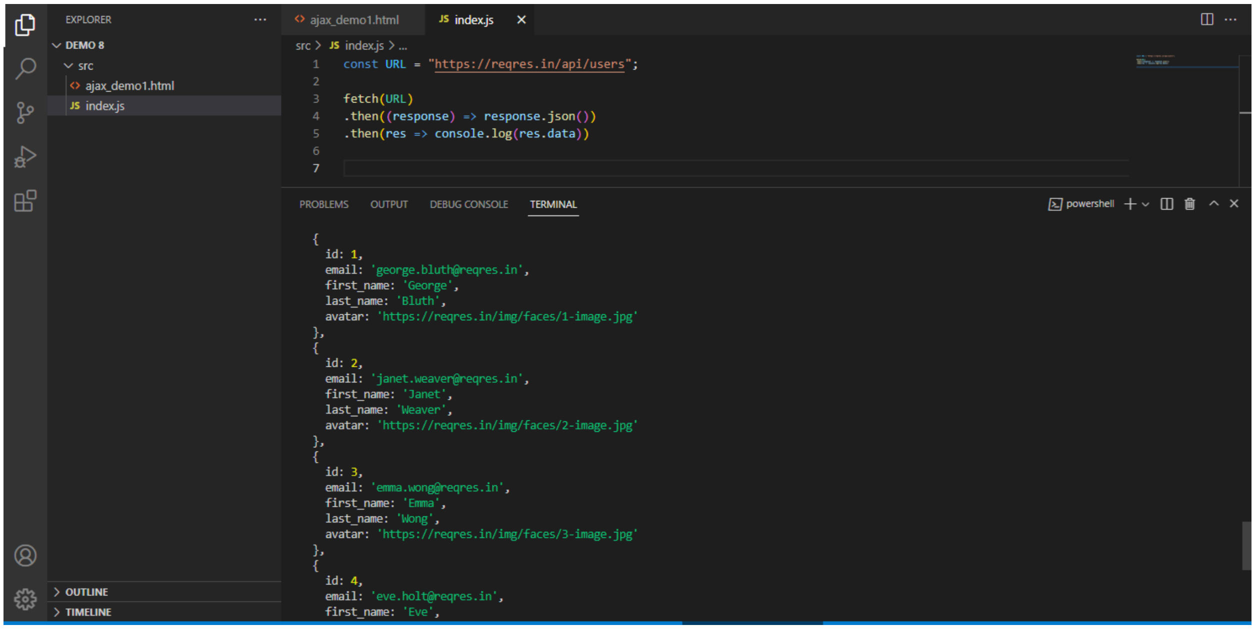
node index.js



```
const URL = "https://reqres.in/api/users";

fetch(URL)
.then((response) => response.json())
.then(res => console.log(res.data))
```

```
Demo for JS\Demo 8\src> node index.js
```

2.5 Click enter and the following data will be shown:



```
const URL = "https://reqres.in/api/users";

fetch(URL)
.then((response) => response.json())
.then(res => console.log(res.data))
```

```
  {
    id: 1,
    email: 'george.bluth@reqres.in',
    first_name: 'George',
    last_name: 'Bluth',
    avatar: 'https://reqres.in/img/faces/1-image.jpg'
  },
  {
    id: 2,
    email: 'janet.weaver@reqres.in',
    first_name: 'Janet',
    last_name: 'Weaver',
    avatar: 'https://reqres.in/img/faces/2-image.jpg'
  },
  {
    id: 3,
    email: 'emma.wong@reqres.in',
    first_name: 'Emma',
    last_name: 'Wong',
    avatar: 'https://reqres.in/img/faces/3-image.jpg'
  },
  {
    id: 4,
    email: 'eve.holt@reqres.in',
    first_name: 'Eve',
```