

| Title: Implementation of Advanced JavaScript Concept |
| --- |

**AIM:** To Implement the Concept of Advanced JavaScript

**Problem Definition:**

-Demonstrate the Concept of Advanced JavaScript With the help of Example.

\*(Students have to perform the task assigned within group and demonstrate the same).

**Resources used:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO 1:**.Build full stack applications in JavaScript using the MERN technologies.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Shelly Powers Learning Node O’ Reilly 2 nd Edition, 2016.

**Pre Lab/ Prior Concepts:**

**Methodology:**

**Implementation Details:**

**//filter unique array numbers**

**<!DOCTYPE *html*>**

**<html *lang*="en">**

**<head>**

**<meta *charset*="UTF-8">**

**<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">**

**<title>Unique Array Members</title>**

**</head>**

**<body>**

**<h1>Filter Unique Array Members</h1>**

**<button *onclick*="showUnique()">Show Unique Array</button>**

**<p *id*="unique-output"></p>**

**<script>**

**function unique(*arr*) {**

**return [...new Set(*arr*)];**

**}**

**function showUnique() {**

**const arr = [1, 2, 2, 3, 4, 4, 5];**

**const uniqueArray = unique(arr);**

**document.getElementById('unique-output').innerText = `Unique Array: ${uniqueArray}`;**

**}**

**</script>**

**</body>**

**</html>**

**//filter anagrams**

**<!DOCTYPE *html*>**

**<html *lang*="en">**

**<head>**

**<meta *charset*="UTF-8">**

**<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">**

**<title>Filter Anagrams</title>**

**</head>**

**<body>**

**<h1>Filter Anagrams</h1>**

**<button *onclick*="showAnagrams()">Show Unique Anagrams</button>**

**<p *id*="anagram-output"></p>**

**<script>**

**function areAnagrams(*word1*, *word2*) {**

**const normalize = *str* => *str*.toLowerCase().split('').sort().join('');**

**return normalize(*word1*) === normalize(*word2*);**

**}**

**function filterAnagrams(*words*) {**

**const uniqueWords = [...new Set(*words*)];**

**const result = [];**

**uniqueWords.forEach(*word* => {**

**if (!result.some(*existingWord* => areAnagrams(*existingWord*, *word*))) {**

**result.push(*word*);**

**}**

**});**

**return result;**

**}**

**function showAnagrams() {**

**const words = ['listen', 'silent', 'enlist', 'hello', 'world'];**

**const filteredAnagrams = filterAnagrams(words);**

**document.getElementById('anagram-output').innerText = `Filtered Anagrams: ${filteredAnagrams}`;**

**}**

**</script>**

**</body>**

**</html>**

**//iterable keys**

**<!DOCTYPE *html*>**

**<html *lang*="en">**

**<head>**

**<meta *charset*="UTF-8">**

**<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">**

**<title>Iterable Keys</title>**

**</head>**

**<body>**

**<h1>Iterable Keys</h1>**

**<button *onclick*="showKeys()">Show Map Keys</button>**

**<p *id*="keys-output"></p>**

**<script>**

**function showKeys() {**

**const map = new Map([**

**['a', 1],**

**['b', 2],**

**['c', 3]**

**]);**

**const keysArray = [...map.keys()];**

**keysArray.push('d');**

**document.getElementById('keys-output').innerText = `Map Keys: ${keysArray}`;**

**}**

**</script>**

**</body>**

**</html>**

**//fetch github users**

**<!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>Fetch GitHub Users</title>**

**</head>**

**<body>**

**<div *class*="container">**

**<h2>GitHub Users</h2>**

**<div *id*="output"></div>**

**</div>**

**<script>**

**async function getUsers(*names*) {**

**const baseUrl = 'https://api.github.com/users/';**

**try {**

**const fetchPromises = *names*.map(*name* =>**

**fetch(`${baseUrl}${*name*}`)**

**.then(*response* => {**

**if (!*response*.ok) {**

**throw new Error(`User ${*name*} not found`);**

**}**

**return *response*.json();**

**})**

**);**

**const users = await Promise.all(fetchPromises);**

**return users;**

**} catch (error) {**

**console.error('Error fetching users:', error);**

**return [];**

**}**

**}**

**const usernames = ['irohitdeshpande'];**

**getUsers(usernames).then(*users* => {**

**const output = document.getElementById('output');**

**output.innerHTML = '<h3>GitHub Users:</h3>' +**

***users*.map(*user* => `**

**<div>**

**<h4>${*user*.login}</h4>**

**<img src="${*user*.avatar\_url}" alt="${*user*.login}" width="100">**

**<p><a href="${*user*.html\_url}" target="\_blank">Profile</a></p>**

**</div>**

**`).join('');**

**});**

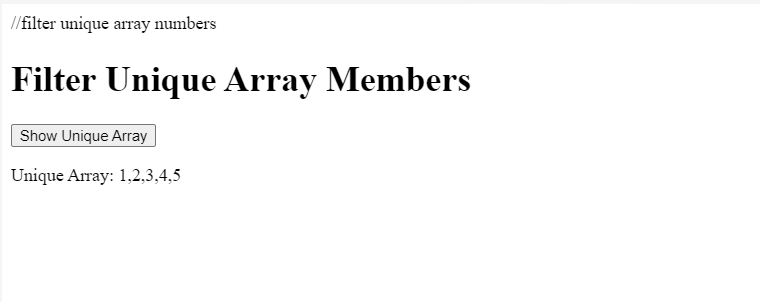
**</script>**

**</body>**

**</html>**

**Steps for execution:**

1. Filter unique numbers



1. Filter anagrams



1. Iterable Keys



1. Fetch github users



**Conclusion:** Demonstrated a few advanced concepts of JavaScript using real-life examples.