

Exno:	API calls using JavaScript
Date:	

**AIM:**

To create a Quote Display App that fetches and displays quotes based on the selected author and user-specified number of quotes.

**DESCRIPTION:**

The API used in this exercise is the **DummyJSON Quotes API** (<https://dummyjson.com/quotes>). This API provides a collection of quotes with information such as the quote text and the author. Each quote has an associated author, and the API allows you to retrieve a list of quotes. The app uses this API to fetch quotes based on the selected author and displays them to the user, ensuring it only shows the requested number of quotes, if available.

**PROCEDURE:**

Tags used	Description
<HTML>	Represents the root of an HTML document.
<head>	Defines a title in the browser toolbar.
<body>	It contains all the contents of an HTML document.
<h1> to <h6>	Used to define the HTML headings.
<p>	Used to define a paragraph.

Javascript keywords	Description
document.getElementById()	<ul style="list-style-type: none"> <li>Accesses the HTML element with the id</li> </ul>
innerHTML	<ul style="list-style-type: none"> <li>a property that allows you to get or set the HTML content inside an element. When used to set content, it interprets any HTML tags as part of the content</li> </ul>
createElement("tag")	<ul style="list-style-type: none"> <li>Creates a new HTML element (e.g., li, button).</li> </ul>
textContent	<ul style="list-style-type: none"> <li>Sets or gets the text inside an element.</li> </ul>
remove()	<ul style="list-style-type: none"> <li>Deletes an element from the DOM</li> </ul>
innerHTML	<ul style="list-style-type: none"> <li>Sets or gets the HTML inside an element.</li> </ul>
event.preventDefault	<ul style="list-style-type: none"> <li>Prevents the default action</li> </ul>

	associated with the event from occurring.
<code>addEventListener</code>	<ul style="list-style-type: none"> <li>Attaches an event handler to a specified element without overwriting existing event handlers.</li> </ul>
<code>Map</code>	<ul style="list-style-type: none"> <li><b>Purpose:</b> Creates a new array populated with the results of calling a provided function on every element in the array.</li> </ul>
<code>await</code>	<ul style="list-style-type: none"> <li><b>Purpose:</b> Pauses the execution of an <code>async</code> function until the <code>Promise</code> is resolved or rejected.</li> </ul>
<code>async</code>	<ul style="list-style-type: none"> <li><b>Purpose:</b> Declares a function as asynchronous, which means it will always return a <code>Promise</code> and allows you to use <code>await</code> inside it.</li> </ul>
<code>alert()</code>	<ul style="list-style-type: none"> <li><b>Purpose:</b> Displays an alert dialog with a specified message.</li> </ul>

**Code:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quotes by Author</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f7f6;
      color: #333;
      margin: 0;
      padding: 0;
      text-align: center;
    }

    h2 {
      color: #2c3e50;
      font-size: 28px;
      margin-top: 50px;
    }

    select, input, button {
      padding: 10px;

```

```
font-size: 16px;
margin: 10px;
border-radius: 5px;
border: 1px solid #ddd;
width: 300px;
}

button {
  background-color: #3498db;
  color: white;
  border: none;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #2980b9;
}

#quote-container {
  margin-top: 30px;
  padding: 20px;
  border: 1px solid #ddd;
  border-radius: 5px;
  background-color: #fff;
  width: 60%;
  display: inline-block;
}

#quote-text {
  font-size: 18px;
  font-style: italic;
  margin-bottom: 15px;
  color: #7f8c8d;
}

#quote-author {
  margin-top: 10px;
  font-weight: bold;
  color: #34495e;
}

select {
  background-color: #ecf0f1;
}

input[type="number"] {
```

```
background-color: #ecf0f1;
}

button:active {
  transform: scale(0.98);
}

#quote-container p {
  font-size: 16px;
  margin: 8px 0;
  color: #2c3e50;
}

</style>
</head>
<body>

<h2>Find Quotes by Author</h2>

<select id="author-select">
  <option value="">Select an author</option>
</select>
<input type="number" id="quote-count" placeholder="Enter number of quotes" min="1" />
<button onclick="fetchQuotesBySelectedAuthor()">Get Quotes</button>

<center><div id="quote-container">
  <p id="quote-text">Select an author and number of quotes to see results.</p>
  <p id="quote-author"></p>
</div></center>

<script>
  let allQuotes = [];
  let uniqueAuthors = [];
  let authorQuotes = [];
  let currentIndex = 0;

  async function fetchAllQuotes() {
    try {
      const response = await fetch("https://dummyjson.com/quotes");
      const data = await response.json();

      allQuotes = data.quotes;

      // Extract unique authors
      uniqueAuthors = [...new Set(allQuotes.map(quote => quote.author))];

      // Populate dropdown
```

```
const authorSelect = document.getElementById("author-select");
uniqueAuthors.forEach(author => {
  let option = document.createElement("option");
  option.value = author;
  option.textContent = author;
  authorSelect.appendChild(option);
});
} catch (error) {
  alert("Error fetching quotes. Please try again.");
}
}

function fetchQuotesBySelectedAuthor() {
  const selectedAuthor = document.getElementById("author-select").value;
  const numberOfQuotes = parseInt(document.getElementById("quote-count").value);

  if (!selectedAuthor) {
    alert("Please select an author!");
    return;
  }

  if (isNaN(numberOfQuotes) || numberOfQuotes < 1) {
    alert("Please enter a valid number of quotes!");
    return;
  }

  authorQuotes = allQuotes.filter(quote => quote.author === selectedAuthor);

  if (numberOfQuotes > authorQuotes.length) {
    alert(`This author only has ${authorQuotes.length} quotes available. Please enter a
number less than or equal to ${authorQuotes.length}.`);
    return;
  }

  const quotesToDisplay = authorQuotes.slice(0, numberOfQuotes);

  currentIndex = 0;
  displayQuotes(quotesToDisplay);
}

function displayQuotes(quotes) {
  if (quotes.length > 0) {
    document.getElementById("quote-container").style.display = "block";
    let quoteText = "";
    quotes.forEach(quote => {
      quoteText += `

"${quote.quote}"



—
${quote.author}

`;
    });
  }
}
```

---

```
    });

    document.getElementById("quote-text").innerHTML = quoteText;
    document.getElementById("quote-author").textContent = `Quotes by
${quotes[0].author}`;
    } else {
        document.getElementById("quote-text").textContent = "No quotes found for this
author.";
    }
}

fetchAllQuotes();
</script>

</body>
</html>
```

**Output:**

**Find Quotes by Author**

Abdul Kalam

3

Get Quotes

*"The Bay of Bengal is hit frequently by cyclones. The months of November and May, in particular, are dangerous in this regard."*  
— Abdul Kalam

*"Thinking is the capital, Enterprise is the way, Hard Work is the solution."*  
— Abdul Kalam

*"While children are struggling to be unique, the world around them is trying all means to make them look like everybody else."*  
— Abdul Kalam

Quotes by Abdul Kalam

Problem Understanding (5)	Implementation & Output (5)	Time Management (5)	Viva (5)	Total (20)

**Result:**

Thus the a Quote Display App that fetches and displays quotes by making API call and it is demonstrated and verified.