**Practical-2:**

1. Create a Jason object and display the same using node.js in the terminal.
2. Create a Node.js script to read and display the content of the external Jason file.
3. Create and display multi-dimensional Jason Arrays, also demonstrate Accessing individual elements of the multi-dimensional Jason Arrays.
4. simple web-based application example using JavaScript and JSON, you can create a basic HTML page with JavaScript to manipulate JSON data. In this example, let's create a simple to-do list application where tasks are stored in a JSON array. Users can add tasks, mark them as completed, and remove them from the list.

**1.** To display the contents of a JSON object in Node.js, you can use the **console.log** function to print the object to the console. Here's a simple example:

File name**: display.js**

// Sample JSON object

const jsonObject = {

name: 'John Doe',

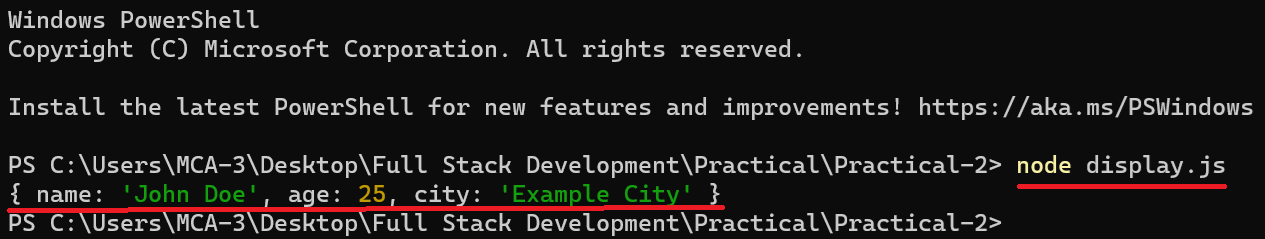
age: 25,

city: 'Example City',

};

// Display JSON object content

console.log(jsonObject);



**2.** File name**: data.json**

{

"name": "John Doe",

"age": 25,

"city": "Example City"

}

File name**: script.js**

const fs = require('fs');

// Specify the path to the JSON file

const filePath = 'C:/Users/MCA-3/Desktop/Full Stack Development/Practical/Practical-2/data.json';

// Read the JSON file

fs.readFile(filePath, 'utf8', (err, data) => {

if (err) {

console.error('Error reading the JSON file:', err);

return;

}

try {

// Parse the JSON data

const jsonObject = JSON.parse(data);

// Display the content

console.log('JSON object content:', jsonObject);

// If you want formatted output

console.log('Formatted JSON object content:', JSON.stringify(jsonObject, null, 2));

} catch (parseError) {

console.error('Error parsing JSON:', parseError);

}

});



**3.** File name**: multi-dimensional-JSON-display.js**

// Create a multi-dimensional JSON array

const multiDimArray = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9],

['apple', 'banana', 'orange']

];

// Display the multi-dimensional JSON array

console.log('Multi-dimensional JSON Array:');

console.log(multiDimArray);

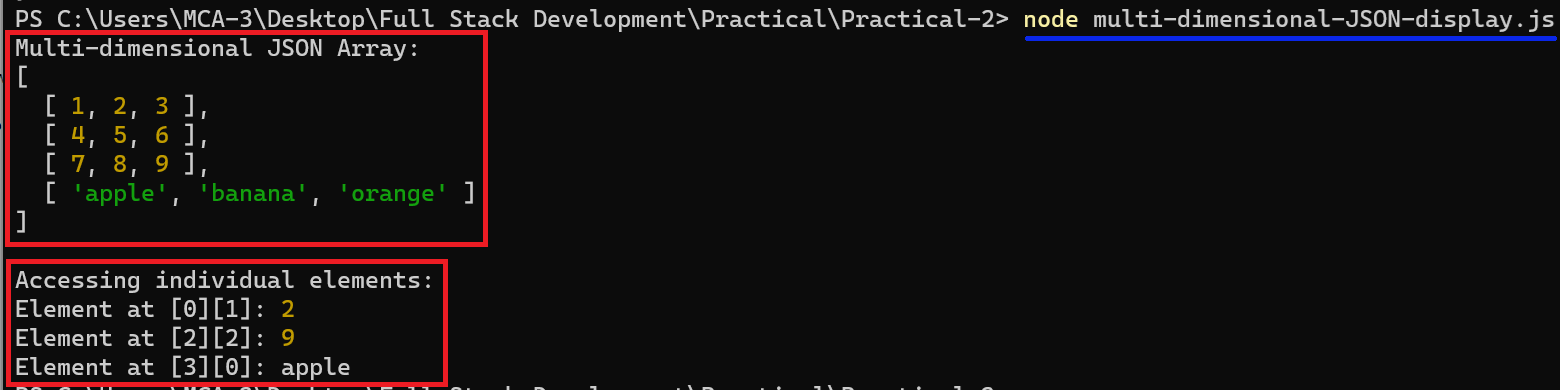
// Accessing individual elements

console.log('\nAccessing individual elements:');

console.log('Element at [0][1]:', multiDimArray[0][1]); // Output: 2

console.log('Element at [2][2]:', multiDimArray[2][2]); // Output: 9

console.log('Element at [3][0]:', multiDimArray[3][0]); // Output: 'apple'



**4.** File name**: manipulated-jasondata.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Todo List App</title>

<style>

ul {

list-style-type: none;

padding: 0;

}

li {

margin: 5px 0;

}

.completed {

text-decoration: line-through;

}

</style>

</head>

<body>

<h1>Todo List App</h1>

<input type="text" id="taskInput" placeholder="Add a new task">

<button onclick="addTask()">Add Task</button>

<ul id="taskList"></ul>

<script>

// Sample JSON array for tasks

let tasks = [

{ id: 1, text: 'Complete assignment', completed: false },

{ id: 2, text: 'Read a book', completed: true },

{ id: 3, text: 'Go for a run', completed: false }

];

// Function to display tasks on the web page

function displayTasks() {

const taskListElement = document.getElementById('taskList');

taskListElement.innerHTML = '';

tasks.forEach(task => {

const li = document.createElement('li');

li.textContent = task.text;

if (task.completed) {

li.classList.add('completed');

}

li.onclick = () => toggleTaskStatus(task.id);

taskListElement.appendChild(li);

});

}

// Function to add a new task

function addTask() {

const taskInput = document.getElementById('taskInput');

const newTaskText = taskInput.value.trim();

if (newTaskText !== '') {

const newTask = {

id: tasks.length + 1,

text: newTaskText,

completed: false

};

tasks.push(newTask);

displayTasks();

taskInput.value = '';

}

}

// Function to toggle the status of a task (completed or not)

function toggleTaskStatus(taskId) {

const taskIndex = tasks.findIndex(task => task.id === taskId);

if (taskIndex !== -1) {

tasks[taskIndex].completed = !tasks[taskIndex].completed;

displayTasks();

}

}

// Initial display of tasks

displayTasks();

</script>

</body>

</html>

