SCTR’s PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

Lab Practice -2 [404184C] : ELECTIVE-III(C) - JavaScript

ACADEMIC YEAR: 2024-25

CLASS	: BE	DIV	: 6	Batch	: P6	DATE	: / /24
Roll No	42130	ABC ID	:			SEMESTER	: I

Experiment No.:

Code:

1. Experiment7.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Custom Array Operations</title>
    <style>
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
            font-family: Arial, sans-serif;
            text-align: center;
        }
        .container {
            border: 1px solid #ddd;
            padding: 20px;
            border-radius: 8px;
            width: 400px;
        }
        button {
            margin: 10px;
        }
        input {
            margin: 5px;
            padding: 5px;
            width: 90%;
        }
        #result {
            margin-top: 20px;
            font-weight: bold;
            white-space: pre-line;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Array Operations</h1>

        <!-- Input for custom array creation -->
        <input type="text" id="customArrayInput" placeholder="Enter array elements separated by commas">
        <button onclick="createCustomArray()">Create Custom Array</button>

        <!-- Input for adding an object with multiple key-value pairs -->
        <input type="text" id="objectInput" placeholder="Enter key-value pairs (e.g., key1:value1, key2:value2)">
        <button onclick="addObject()">Add Object (Multiple Key-Value Pairs)</button>

        <!-- Input for appending new array -->
        <input type="text" id="newArrayInput" placeholder="Enter new array elements separated by commas">
        <button onclick="appendArray()">Append New Array</button>

        <button onclick="checkElementTypes()">Check Element Types</button>

        <div id="result"></div>
    </div>
    <script src="exp7.js"></script>
</body>
</html>
```

2. Experiment7.js:

```
let array = [];

function createCustomArray() {
    const userInput = document.getElementById('customArrayInput').value;
    if (userInput) {
        array = userInput.split(',').map(item => {
            item = item.trim();
            if (!isNaN(item)) return Number(item);  // Convert to number if applicable
            if (item.toLowerCase() === 'true') return true;
            if (item.toLowerCase() === 'false') return false;
            return item;  // Keep as string otherwise
        });
        displayResult(`Custom Array:\n[${array.join(', ')}]`);
    }
}
```

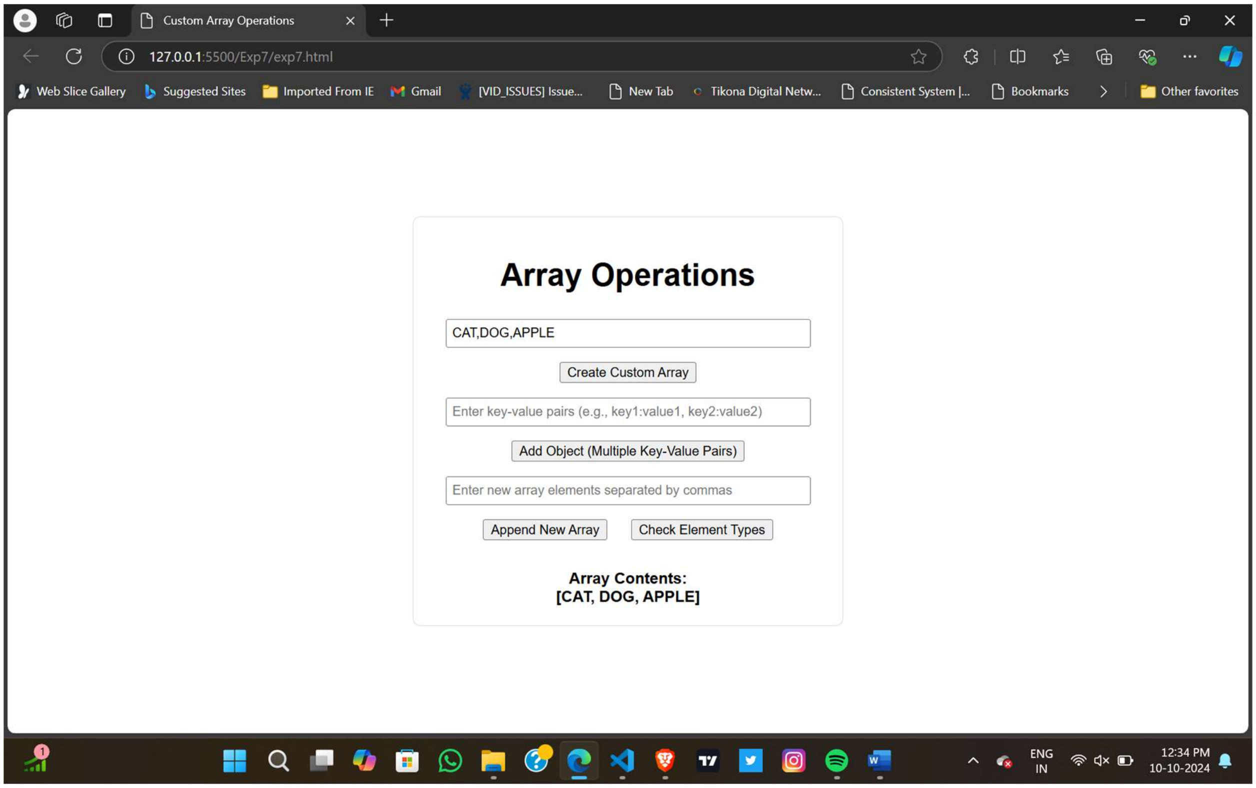
```
function addObject() {
    const userInput = document.getElementById('objectInput').value.trim();
    if (userInput) {
        let obj = {};
        const keyValuePairs = userInput.split(',');
        keyValuePairs.forEach(pair => {
            const [key, value] = pair.split(':').map(item => item.trim());
            if (key && value !== undefined) {
                let parsedValue;
                if (!isNaN(value)) {
                    parsedValue = Number(value);
                } else if (value.toLowerCase() === 'true') {
                    parsedValue = true;
                } else if (value.toLowerCase() === 'false') {
                    parsedValue = false;
                } else {
                    parsedValue = value;
                }
                obj[key] = parsedValue;
            }
        });
        array.push(obj);
        displayResult(`Array after adding object:\n[${array.join(', ')}]`);
    }
}

function appendArray() {
    const userInput = document.getElementById('newArrayInput').value;
    if (userInput) {
        const newArray = userInput.split(',').map(item => {
            item = item.trim();
            if (!isNaN(item)) return Number(item);
            if (item.toLowerCase() === 'true') return true;
            if (item.toLowerCase() === 'false') return false;
            return item;
        });
        array.push(newArray);
        displayResult(`Array after appending new array:\n[${array.join(', ')}]`);
    }
}

function checkElementTypes() {
    let types = array.map(element => {
        if (Array.isArray(element)) {
            return `Array: ${JSON.stringify(element)}`;
        } else if (typeof element === 'object') {
            return `Object: ${JSON.stringify(element)}`;
        } else {
            return `${typeof element}: ${JSON.stringify(element)}`;
        }
    });
    document.getElementById("result").innerHTML = `Element
Types:\n${types.join('\n')}`;
}


function displayResult() {
    const formattedArray = array.map(element => {
        if (typeof element === 'object') {
            return JSON.stringify(element);
        }
        return element;
    });
    document.getElementById("result").innerHTML = `Array
Contents:\n[${formattedArray.join(', ')}]`;
}
```

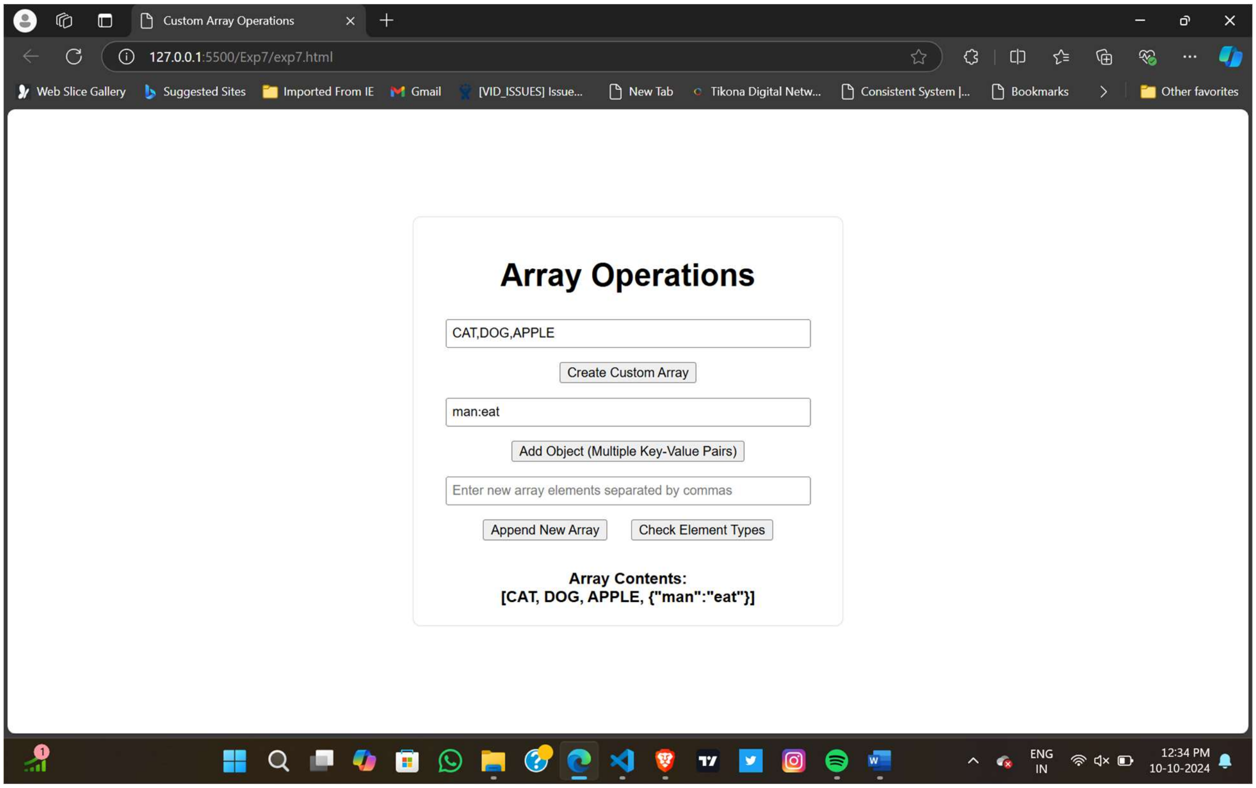
Output:



Array Operations

CAT,DOG,APPLE

Array Contents:
[CAT, DOG, APPLE]

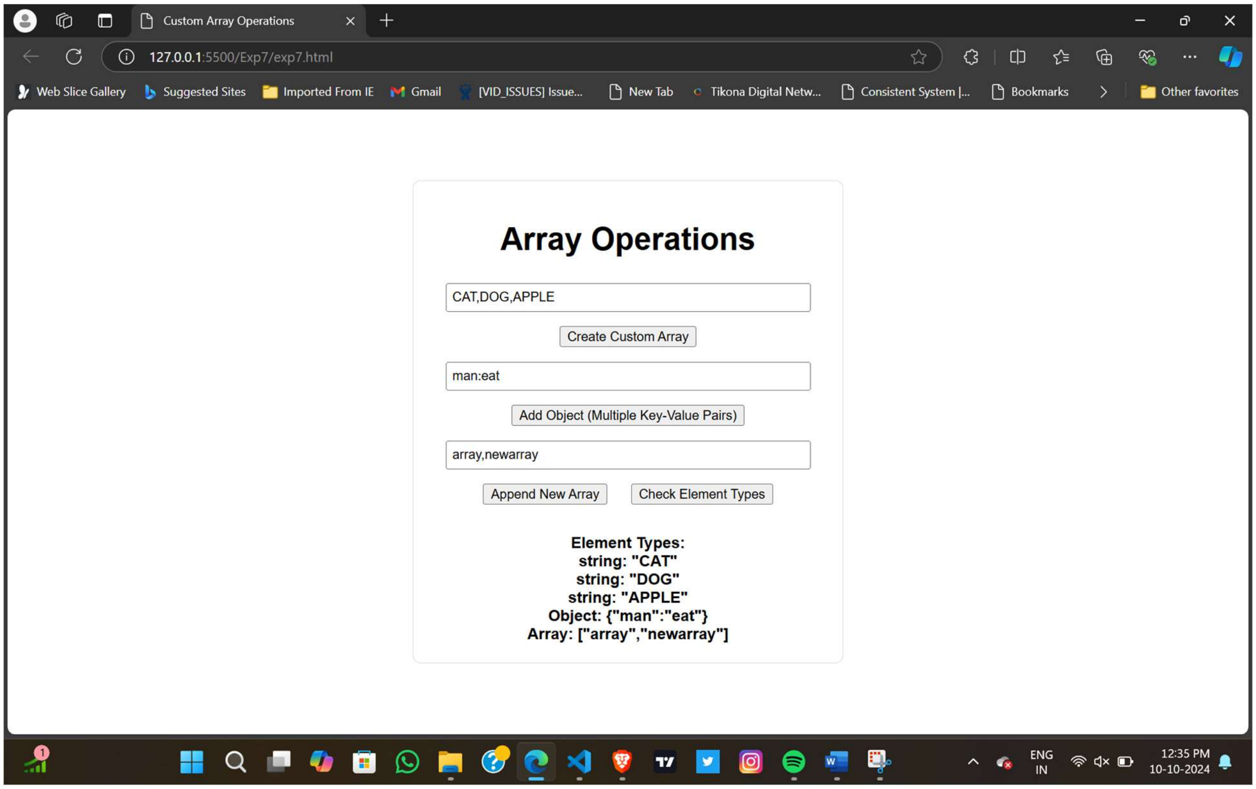


Array Operations

CAT,DOG,APPLE

man:eat

Array Contents:
[CAT, DOG, APPLE, {"man":"eat"}]



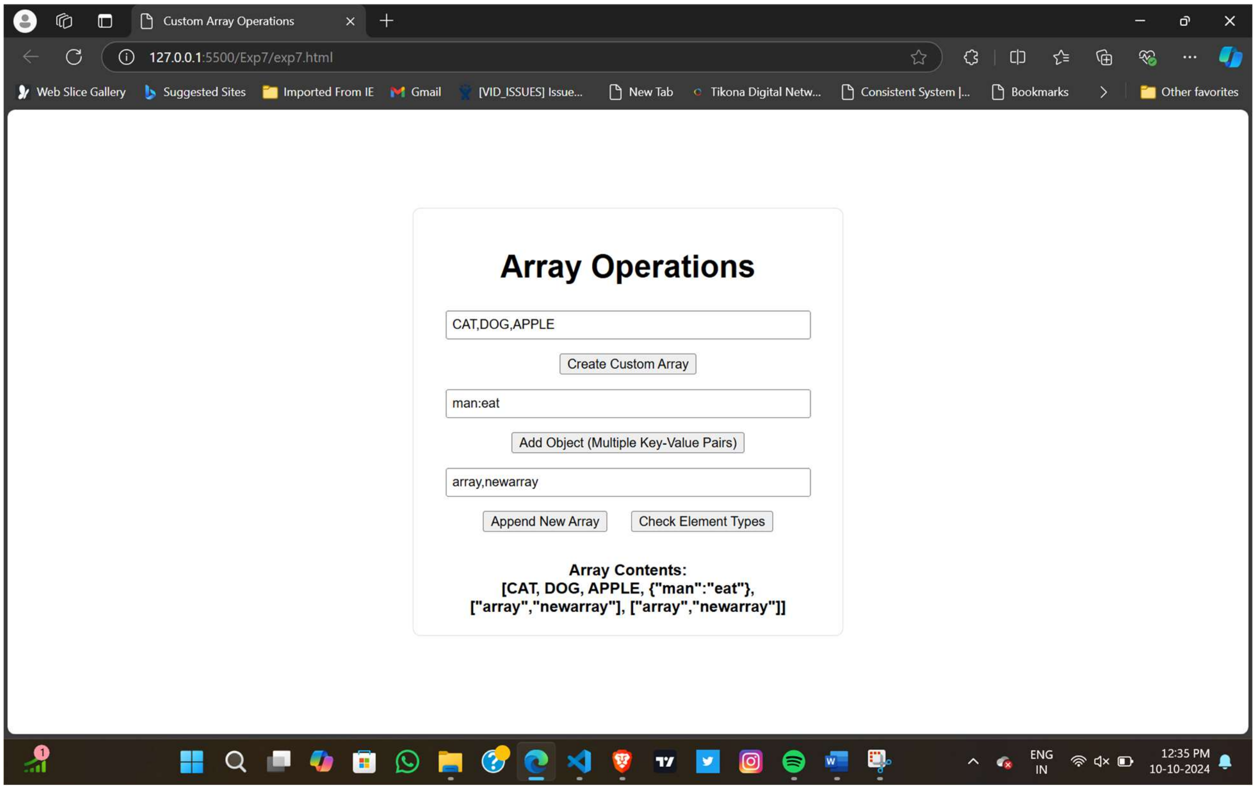
Array Operations

CAT,DOG,APPLE

man:eat

array,newarray

```
Element Types:
string: "CAT"
string: "DOG"
string: "APPLE"
Object: {"man":"eat"}
Array: ["array","newarray"]
```



Array Operations

CAT,DOG,APPLE

man:eat

array,newarray

```
Array Contents:
[CAT, DOG, APPLE, {"man":"eat"},
["array","newarray"], ["array","newarray"]]
```

Date: Course Teacher Sign