JavaScript Lab - 5

Core Concepts in Practice

Course: Introduction to JavaScript
Lab: 05 - JavaScript Commands
Topic: Applying class, Set, Map, and `this`.
Total Marks: 10
Due Date: 4th September, 2025: 11:59pm

Objective

By the end of this lab, you will build a functional, console-based application.

Prerequisites

- A basic understanding of JavaScript variables, functions, and objects.

Building the System (10 Marks)

Follow the steps below to build your event registration system. You can write and test your code in an online editor like CodePen.io as you work.

Step 1: The Attendee (1 Mark)
Task: Create an `Attendee` class with a `constructor` that accepts `name` and `email` and assigns them to the instance.

Step 2: The Event (1.5 Marks)
Task: Create an `Event` class.

- The `constructor` should accept `eventName` and `date`.
- It must have a property `registeredAttendees`, initialized as a `new Set()`.
- It must have a method `register(attendee)` that adds the `attendee.email` to the Set and returns `true` for a new registration or `false` for a duplicate.

Step 3: The EventManager (1.5 Marks)
Task: Create an `EventManager` object.

- It must have a property `events`, initialized as a `new Map()`.
- It must have a method `createEvent(event)` that adds an event to the map.
- It must have the following `getEventSummary` method.

```
getEventSummary: function(eventName) {
    if (!this || !this.events) {
        return "ERROR:INVALID_CONTEXT";
    }
    const event = this.events.get(eventName);
    if (event) {
        return `SUMMARY:${event.eventName}/${event.registeredAttendees.size}`;
    }
    return `ERROR:NOT_FOUND`;
}
```

Step 4: System Assembly (6 Marks)
Task: Write the code to assemble and run your system.

a) Create two `Event` instances: `jsConf` for `'JS Conference'` on `'2025-12-01'` and `pythonSummit` for `'Python Summit'` on `'2025-12-15'`.

b) Add both events to the `EventManager`.

c) Create two `Attendee` instances for 'Hiya Bhatt' (`hiya@example.com`) and 'Kunal Bhosikar' (`kunal@example.com`).

d) Register Hiya and Kunal for `jsConf`.

e) Attempt to register Hiya for `jsConf` a second time.

f) Register Kunal for `pythonSummit`.

Validation and Submission

After completing your code, follow these steps to generate your verification string and prepare your final submission.

1. **Open the Validator Tool:** Navigate to the following link: https://ssd-lab5.netlify.app/

2. **Enter Your Details:** Enter your full Roll Number and official IIIT email address.

3. **Validate Your Code:** Copy your entire JavaScript code and paste it into the text area. Click the "Validate" button.

4. **Get Your String:** If your code is correct, the tool will generate a unique *Verification String*. Copy this entire string.

5. **Prepare Submission Files:**

   - Create a plain text file named `Verify.txt`. Paste your copied *Verification String* into this file.
   - Save your complete JavaScript code in a file named `JS_code.js`.

6. **Create Zip Archive:** Place both files (`Verify.txt` and `JS_code.js`) into a single zip archive named `Rollnumber_Lab5.zip` (e.g., `2024701001_Lab5.zip`).

7. **Submit:** Upload the final `.zip` file to the course portal.

Important Note on Submission

If for any technical reason you are unable to access the validator or generate the Verification String, you may still submit your `.zip` file with only the `JS_code.js` file inside for partial credit.

You must ensure your code adheres strictly to the variable, method, and property names exactly as they are specified in this document (e.g., `eventName`, `createEvent`, etc.).