### **✅ Key Areas of Focus**

1. **Active Student Engagement**
   * Students aren't just passively learning; they’re actively teaching and learning from peers. That’s huge for retention and agency.
2. **Rotational Involvement**
   * Having a new group present each time ensures inclusivity and gives everyone a turn to shine.
3. **Collaboration Emphasis**
   * Requiring all group members to contribute prevents free-riding and builds teamwork skills.
4. **Micro-Seminar Format (5–10 min)**
   * Time-efficient, keeps the energy high, and doesn't overwhelm the group or audience.
5. **Technical Topic Focus**
   * Aligns perfectly with web development teaching goals and ensures exposure to diverse subtopics.

### **💡 Guidelines**

1. **Clear Rubric / Expectations**
   * Have a simple evaluation or feedback rubric that encourages clarity, accuracy, presentation skills, and team balance.

| **Objective** | **Evaluation Criteria** | **1 - Beginning** | **2 - Developing** | **3 - Proficient** | **4 - Exemplary** |
| --- | --- | --- | --- | --- | --- |
| **Boost Confidence** | Maintains eye contact | Avoids eye contact | Occasional eye contact | Consistent with most audience | Confident, steady, across audience |
|  | Audible and clear voice | Very soft or mumbled | Sometimes unclear | Mostly clear and audible | Clear, confident, well-paced |
|  | Body language & presence | Closed or stiff | Somewhat open | Open and engaged | Expressive and natural |
|  | Engages with audience | Reads slides only | Minimal engagement | Asks 1–2 questions or invites input | Actively interacts (e.g., humor, polls, Q&A) |
| **Foster Collaboration** | Equal participation | 1–2 students dominate | Some students silent | All students contribute, unevenly | All students actively participate |
|  | Group coordination | Disorganized, no flow | Some flow but disconnected | Mostly well-coordinated | Seamless transitions and teamwork |
| **Develop Communication** | Clarity of explanation | Unclear or confusing | Partly understandable | Clear with minor gaps | Very clear, structured, and accessible |
|  | Use of examples | No examples used | Vague or irrelevant | Relevant, understandable | Creative, relatable examples |
|  | Visual aids | None or distracting | Basic or unclear | Clear and supportive | Visually engaging and enhances content |
| **Deepen Subject Knowledge** | Accuracy of content | Many errors | Some errors | Mostly accurate | Fully accurate and precise |
|  | Response to questions | Cannot answer | Struggles or vague | Adequate answers | Confident and detailed responses |

1. **Roles Within Groups** (Optional)  
   * To avoid confusion or uneven participation, assign rotating roles like:  
     + Researcher
     + Presenter
     + Visual Aids Lead
     + Summarizer

This helps students understand the value of different contributions.

1. **Use of Visual Aids or Demos**
   * Encourage use of slides, whiteboard, or quick live coding/demo. Helps visual learners and boosts engagement.
2. **Peer Feedback (2 mins after each session) (Optional)**
   * Let the rest of the class share one takeaway and one suggestion. This encourages active listening and gives presenters a growth moment.
3. **Topic Difficulty Tuning**
   * Ensure topics are at an appropriate difficulty level based on what you’ve already covered or are about to cover.
4. **Rotation Log or Calendar**
   * Keep a simple tracker so students know when their group is up next. This increases preparedness.

### **1. Presentation Outline**

A basic structure students can follow when preparing their talk.

\*\*Topic Title:\*\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. \*\*Introduction (1 min)\*\*

- Who are we?

- What are we going to talk about?

2. \*\*Main Concepts (3–5 mins)\*\*

- Key Point 1:

- Explanation

- Example/demo

- Key Point 2:

- Explanation

- Example/demo

3. \*\*Real-World Application (1 min)\*\*

- Where is this used in real-life tech?

4. \*\*Wrap-Up (1 min)\*\*

- Summary of key takeaways

- Questions (if any)

### **2. Group Role Assignment**

Helps students divide work fairly and ensures each person contributes.

| Name | Role | Responsibility |

|-------------|--------------|-------------------------------------------|

| Student A | Researcher | Gathers information and examples |

| Student B | Presenter | Delivers the talk |

| Student C | Visual Lead | Prepares slides or demos |

| Student D | Moderator | Introduces topic, wraps up, handles Q&A |

### **3. Peer Feedback Form (Optional)**

Used by the audience after the presentation to give quick feedback.

\*\*Group Name / Topic: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*\*

1. One thing I learned: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. One suggestion for improvement: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Clarity (1–5): \_\_\_ | Confidence (1–5): \_\_\_ | Teamwork (1–5): \_\_\_

| Curriculum topics | General tech |
| --- | --- |
| What is the DOM and Why Does It Matter in Web Development?  Explain what is a virtual DOM used by React. | What is GitHub and Why Do Developers Use It? Explain the importance of version control. |
| Selecting and Manipulating DOM Elements Using JavaScript  Compare different DOM querying methods (getElementById vs querySelector), and What are **reflow** and **repaint** operations? | What are the steps between typing a URL and seeing a webpage?(Browser to Server Journey) |
| Handling Events in JavaScript: click, change, mouseover, and more  What are events in JS? Common examples? How to add event listeners? Types of event listeners? Event handling best practices? Discuss what is event bubbling and event capturing in JS. | Careers in Web Development: What Roles Exist in the Industry? |
|  | What is an API and How Does It Work? |
| Common Mistakes in Event Handling and How to Avoid Them | What is HTTP and What Do Status Codes Like 404, 200, or 500 Mean? - Introduce HTTP request/response cycle + common codes. |
| What is Asynchronous Programming and Why is it Needed in JavaScript?  Discuss the difference between **concurrency** and **parallelism**. While JavaScript runs **concurrently** using the event loop, introduce concepts like **Web Workers** for parallel execution in JavaScript and how they differ from asynchronous code. | How Does a Browser Render a Web Page? |
| Understanding Callbacks in JavaScript with Real-Life Analogies.  Discuss call back hell and how to avoid it. **Refactor** a nested-callback example into named, modular functions. | What are Cookies, LocalStorage, and Sessions? - Introduce storage, persistence, and session management. |
| Promises in JavaScript: Syntax, Chaining, and Real-World Use  **Compare** the use of Promises to callbacks and discuss the **advantages** of using Promises over callback functions. | What is SEO and Why Should Web Developers Care About It? - Discuss page structure, metadata, and accessibility. |
| Using async and await to Simplify Asynchronous Code  Discuss error Handling with Async/Await, advantages of Async/Await over promises. Research how **async iterators** work. | What is Accessibility (a11y) in Web Development? |