Here's your **ultra-detailed Day 3 schedule** focused on JavaScript fundamentals, following the same rigorous format:

🗓️ Day 3: JavaScript Fundamentals

⏰ **7 Hours** (9:00 AM - 4:30 PM)
🎯 **Goal**: Build an interactive task manager with DOM manipulation

---

🌞 Morning Session (9:00 AM - 12:30 PM)

1. 🧊 Warmup & Review (9:00-9:20 AM)

What to Say:
"HTML/CSS structure websites, but JavaScript makes them interactive. Today we bring pages to life!"

Activities:

- **9:00-9:10**: Fix this broken Bootstrap navbar (review):

```
<nav class="navbar">
  <!-- Missing navbar-expand class -->
  <div class="container-fluid">
    <a class="navbar-brand" href="#">My Site</a>
  </div>
</nav>
```

- **9:10-9:20**: JS Setup in HTML:

```html
<!DOCTYPE html>
<html>
    <body>
      <script>
        // Our JS code goes here
        alert("Hello World!");
      </script>
    </body>
</html>
```

2. 📜 JavaScript Basics (9:20-10:20 AM)

What to Say:
"JS is the brain of web development. Let's learn its core vocabulary."

Content:

- **9:20-9:35**: Variables & Data Types

```javascript
// String
let userName = "Alice";

// Number
const age = 25;

// Boolean
const isStudent = true;

// Array
const skills = ["HTML", "CSS"];

// Object
const user = {
  name: "Bob",
  age: 30
};
```

- **9:35-9:50**: Operators & Template Literals

```
// Arithmetic
let sum = 10 + 5;

// Comparison
const isAdult = age >= 18;

// String interpolation
console.log(`Hello ${userName}! You are ${age} years old.`
);
```

- **9:50-10:20**: Lab - User Profile Generator

```
const user = {
  name: prompt("Enter your name"),
  age: parseInt(prompt("Enter your age"))
};
document.write(`<h1>Welcome ${user.name}!</h1>`);
```

☕ 10:20-10:35 AM | 🍵 Break

---

3. 🛠️ Functions & Conditionals (10:35 AM - 12:30 PM)

What to Say:
"Functions are reusable code blocks. Conditionals make decisions."

Content:

- **10:35-10:50**: Function Declaration

```
function greet(name) {
  return `Hello ${name}!`;
}
greet("Alice"); // "Hello Alice!"
```

- **10:50-11:05**: Arrow Functions

```
const add = (a, b) => a + b;
console.log(add(2, 3)); // 5
```

- **11:05-11:20**: If/Else Statements

```
const age = 20;
if (age >= 18) {
  console.log("Adult");
} else {
  console.log("Minor");
}
```

- **11:20-12:30**: Project - Age Checker App

```
<script>
  function checkAge() {
    const age = document.getElementById("ageInput").value;
    const result = age >= 21 ? "Cheers!" : "Too young!";
    document.getElementById("result").textContent = result;
  }
</script>
<input id="ageInput" type="number">
<button onclick="checkAge()">Check</button>
<p id="result"></p>
```

🌇 Afternoon Session (1:00 PM - 4:30 PM)

🍱 12:30-1:00 PM | 🍜 Lunch

---

4. 🖱️ DOM Manipulation (1:00-2:00 PM)

What to Say:
"The DOM is JavaScript's window into HTML. Let's learn to control it."

Content:

- **1:00-1:20**: Selecting Elements

```
// By ID
const header = document.getElementById("header");

// By Class
const items = document.getElementsByClassName("item");

// Query Selector
const button = document.querySelector(".btn-primary");
```

- **1:20-1:40**: Modifying Content

```
// Change text
header.textContent = "New Title";

// Change HTML
header.innerHTML = "<em>New</em> Title";

// Change styles
header.style.color = "blue";
```

- **1:40-2:00**: Lab - Dark Mode Toggle

```
function toggleDarkMode() {
  document.body.classList.toggle("dark-mode");
}
```

5. 🎯 Event Listeners (2:00-2:30 PM)

What to Say:
"Events trigger actions. Let's make buttons actually work!"

Content:

- **2:00-2:15**: Click Events

```
document.getElementById("myBtn").addEventListener("click", () => {
  alert("Button clicked!");
});
```

- **2:15-2:30**: Form Submission

```
document.querySelector("form").addEventListener("submit", (e) => {
  e.preventDefault(); // Stop page reload
  console.log("Form submitted!");
});
```

☕ 2:30-2:45 PM | 🍵 Break

6. 📝 Project: Task Manager (2:45-4:15 PM)

What to Say:
"Let's combine everything into a real app."

Step-by-Step:

- **2:45-3:15**: HTML Structure

```
<div id="app">
  <input id="taskInput" placeholder="New task">
  <button id="addBtn">Add</button>
  <ul id="taskList"></ul>
</div>
```

- **3:15-3:45**: JavaScript Functionality

```
const tasks = [];

function addTask() {
  const input = document.getElementById("taskInput");
  tasks.push(input.value);
  renderTasks();
}

function renderTasks() {
  const list = document.getElementById("taskList");
  list.innerHTML = tasks.map(task => `<li>${task}</li>`).join("");
}
```

- **3:45-4:15**: Debugging Session

```
// Broken Code to Fix:
document.getElementById("addBtn").addEventListner("click", addTask);
// ^^ Typo in addEventListener
```

7. 🎤 Presentations (4:15-4:30 PM)

Each student demonstrates:

1. One JavaScript feature they implemented
2. One bug they fixed

📌 Key Examples

DOM Manipulation Cheatsheet:

```
// Create element
const newDiv = document.createElement("div");

// Add class
newDiv.classList.add("alert");

// Append to DOM
document.body.appendChild(newDiv);

// Remove element
newDiv.remove();
```

Event Listener Types:

```
// Mouse events
element.addEventListener("mouseover", doSomething);

// Keyboard events
document.addEventListener("keydown", (e) => {
  if (e.key === "Enter") submitForm();
});

// Window events
window.addEventListener("resize", handleResize);
```

Complete Task Manager Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Task Manager</title>
    <style>
      .completed { text-decoration: line-through; }
    </style>
  </head>
  <body>
    <div id="app">
      <input id="taskInput" placeholder="New task">
      <button id="addBtn">Add Task</button>
      <ul id="taskList"></ul>
    </div>

    <script>
      const tasks = [];

      document.getElementById("addBtn").addEventListener("click", () => {
        const input = document.getElementById("taskInput");
        if (input.value.trim() === "") return;

        tasks.push({
          text: input.value,
          completed: false
        });

        input.value = "";
        renderTasks();
      });

      function renderTasks() {
        const list = document.getElementById("taskList");
        list.innerHTML = tasks.map((task, index) => `
          <li class="${task.completed ? 'completed' : ''}">
            ${task.text}
            <button onclick="toggleTask(${index})">Toggle</button>
          </li>
        `).join("");
      }

      function toggleTask(index) {
        tasks[index].completed = !tasks[index].completed;
        renderTasks();
      }
    </script>
  </body>
</html>
```

This schedule delivers:

- ✅ 15-30 minute focused segments
- ✅ 50+ practical JavaScript examples
- ✅ Complete project build-along
- ✅ Common error debugging practice
- ✅ Gradual complexity progression