Lesson 6: Multi-View UI (Page Switching)

🎯 Goal

Students will learn how to create simple multi-view apps by showing and hiding different sections of a webpage using JavaScript. This simulates switching between pages like in real applications.

⏱️ Total Duration: 90–120 minutes

Level: Intermediate

🧭 Lesson Outline

1. Warm-Up (10 minutes)

- Ask students:
  - "Have you seen apps with different pages like Home, Settings, or Profile?"
  - "What if we want to build that without loading a new page?"
- Demo a simple UI where clicking buttons switches content.

2. Basic Concept: Show/Hide Views (20 minutes)

- ✅ Use CSS `display` property to toggle sections
- 📖 Example:

```
document.getElementById("view1").style.display = "none";
document.getElementById("view2").style.display = "block";
```

- 🧪 Student Practice:
  - Toggle between two `<div>`s by clicking a button

3. Structure a Multi-View App (20 minutes)

- Layout with 3 views: Home, Notes, Settings
- Navigation buttons at the top
- Keep each view in a separate `<div>`

💡 Navigation Code Example:

```
function showView(id) {
  let views = ["homeView", "notesView", "settingsView"];
  views.forEach(v => {
    document.getElementById(v).style.display = (v === id) ? "block" : "none";
  });
}
```

---

4. Mini Project: Build Your Own Multi-View App

Create a small **multi-view web app** using HTML, CSS, and JavaScript. You'll use `display` toggling and DOM manipulation to simulate switching between different pages — just like in real apps (e.g., Instagram or Notion).

Your app must include the following **three views**:

1. **Home View**
    - Shows a welcome message
    - Includes navigation buttons to switch to other views
2. **Todo View**
    - Has a form to add todo items (an input and a "+" button)
    - Shows the list of added todos
    - Stores todos using `localStorage`
3. **Fun Fact View**

  - When this view is opened, display a random fun fact (you can hardcode 3–5 facts in an array)

Technical Requirements

- Use only **one HTML page**
- Use **three `<div>`s**, one for each view, and show/hide using `style.display` or CSS classes
- Use JavaScript to:
  - Handle navigation button clicks
  - Update the content dynamically (e.g., show todos, show random fun fact)
  - Save and retrieve todos from `localStorage`

Bonus (Optional Challenges)

- ✅ Add a "Delete" button next to each todo item
- ✅ Use `classList` and CSS transitions to make views fade in/out
- ✅ Add a simple settings page to let users change the background color

5. Extension Challenge (Optional/Homework)

- Add delete note feature
- Allow editing notes
- Add view transitions (e.g. fade in/out)

6. Recap & Q&A (10 minutes)

- What’s the benefit of switching views on one page?
- How can we organize our app logic by view?
- How does this connect to real apps like Instagram or Notion?

✅ Key Takeaways

- `display: none` and `block` control visibility
- Apps don’t always need multiple HTML pages
- View-based design helps structure more complex apps

Next lesson: we’ll explore **storing & editing dynamic data**, like building a full task manager or to-do app! ✅