JavaScript Training Program (1 Week)

Goal

Take a fresher from **basic JS** → **advanced ES6+** → **real-world development**, ending with a **fully functional project**.

Day 1 – Core JS Foundations

Topics to Learn (Why Important)

- Variables: `var`, `let`, `const` (scope differences).
- Data Types: string, number, boolean, null, undefined, object, symbol.
- String methods: `.toUpperCase()`, `.toLowerCase()`, `.slice()`, `.includes()`, `.replace()`.
- Number methods: `toFixed()`, `Math.random()`, `Math.floor()`, `parseInt()`.
- Operators & Control flow (`if`, `else`, `switch`, loops).

Task: "User Profile Generator"

- Ask for `name`, `age`, `email` via prompt/CLI.
- Format strings (capitalize name, validate email).
- Generate a unique **userID** using `Math.random()`.
- Show profile summary.

Why: Covers **strings, numbers, validation** → used in every signup/login system.

Day 2 – Functions, Arrays & Data Processing

Topics

- Functions (declaration, arrow functions).
- Default/rest parameters.
- Arrays: `map`, `filter`, `reduce`, `find`, `some`, `every`.

- Spread & rest operators.

Task: “User Management System – Part 1”

- Store multiple users in an **array of objects**.
- Implement functions:
  - `addUser()`
  - `getUserById()`
  - `getAdults()` (using filter)
  - `getAverageAge()` (using reduce)

Why: Prepares them for handling **real user data in arrays**.
Connected to Day 1: Uses user profiles created earlier.

Day 3 – Objects, DOM Manipulation & Events

Topics

- Objects & nested objects.
- Object methods, destructuring.
- DOM selection & manipulation (`querySelector`, `.textContent`, `.classList`).
- Event listeners (`click`, `input`, `submit`).

Task: “Interactive User Dashboard – Part 2”

- Display user list (from Day 2) on a webpage.
- Add button: **Add User** → form input.
- Add button: **Delete User** → remove user from DOM & array.
- Add button: **Highlight Adults** → apply CSS class.

Why: Teaches **DOM + objects + events** in real-world UI.
Connected to Day 2: Works on same user data.

Day 4 – Asynchronous JS (Callbacks, Promises, Async/Await)

Topics

- Sync vs async execution.

- Callbacks → Promises → Async/Await.
- Fetch API basics.
- Error handling with try/catch.

Task: "User Dashboard with API – Part 3"

- Fetch dummy users from API (`https://jsonplaceholder.typicode.com/users`).
- Merge API users with local users (from Day 3).
- Add **Refresh Button** → refetch data.
- Show error message if API fails.

Why: Real-world apps **always fetch from APIs**.
Connected to Day 3: Enhances dashboard with external users.

Day 5 – ES6+ Features, Classes, Modules

Topics

- Template literals.
- Destructuring, spread/rest.
- Classes & inheritance.
- Modules: `import`, `export`.
- Sets & Maps (unique data, key-value storage).

Task: "Advanced User Manager – Part 4"

- Create `User` class → properties: id, name, email, age.
- Add methods: `isAdult()`, `getEmailDomain()`.
- Store users in a **Map** (key = userId).
- Maintain a **Set of unique emails**.

Why: Real projects need **OOP + unique constraints**.
Connected to Day 4: Refactors dashboard using **classes + ES6 features**.

Day 6 – Local Storage, JSON, Error Handling

Topics

- JSON stringify & parse.
- LocalStorage & SessionStorage.
- Error handling: try/catch/finally, custom errors.

Task: "Persistent User Dashboard – Part 5"

- Save user data in localStorage.
- On reload → load users from storage.
- Add error handling (e.g., if JSON parse fails).

Why: Teaches **data persistence** in frontend apps.
Connected to Day 5: Makes dashboard **persistent across reloads**.

Day 7 – Advanced Topics & Final Project

Topics

- Event delegation.
- Debouncing & throttling.
- Advanced ES6: optional chaining, nullish coalescing.
- Modular project structure.

Final Task: "Mini E-Commerce App"

- Display product list (dummy JSON or API).

- Add product to cart (stored in localStorage).
- Show cart summary (items + total).
- Add **debounced search bar** to filter products.
- Use classes (`Product`, `Cart`).
- Use Sets for unique cart items.
- Use Maps for product stock.

Why: Covers **all core-to-advanced JS concepts in one project**.

Connected to All Days:

- Strings & numbers (Day 1) for product names & prices.
- Arrays & functions (Day 2) for cart calculations.
- DOM & events (Day 3) for UI.
- API & async (Day 4) for product data.
- Classes, Sets, Maps (Day 5) for structure.
- LocalStorage (Day 6) for persistence.
- Debouncing & delegation (Day 7) for performance.
