**Seminarul 4 — Partea 3 (Proiecte/teme, extins)**

\*\*Obiectiv.\*\* Această parte conține 45 de proiecte (15 × L1, 15 × L2, 15 × L3) care consolidează cunoștințele despre ES modules, DOM și modelul de evenimente, cu accent pe dependency management, performanță și testabilitate. Pentru fiecare proiect sunt livrate: (a) schelet complet runnable (HTML/CSS/JS), (b) teste unitare în oglindă (Vitest/Jest), (c) README cu learning goals, (d) pentru subsetul L3: teste E2E (Playwright) și PWA (Workbox).

### L1 — Proiecte (15)

\*\*01. L1-p01-esm-bootstrap — Bootstrap ESM: <script type="module">\*\*

\*Specificație.\* Initializează aplicația cu ES modules; conectează /src/main.js din /public/index.html și exportă bootstrap(document).

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p01-esm-bootstrap/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p01-esm-bootstrap/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*02. L1-p02-dom-helpers — DOM helpers cu invariante (qs/qsa)\*\*

\*Specificație.\* Expune utilitare qs/qsa care aruncă pentru lipsă selector și returnează array pentru liste.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p02-dom-helpers/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p02-dom-helpers/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*03. L1-p03-delegation-click — Delegare evenimente pe listă\*\*

\*Specificație.\* Atașează un singur listener pe #list și folosește .closest('li.card') pentru click.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p03-delegation-click/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p03-delegation-click/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*04. L1-p04-once-passive-capture — Opțiuni addEventListener (once/passive/capture)\*\*

\*Specificație.\* Demonstrează {once:true}, {passive:true}, {capture:true} pe butoane dedicate.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p04-once-passive-capture/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p04-once-passive-capture/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*05. L1-p05-custom-event-basics — CustomEvent: emit/ascultă\*\*

\*Specificație.\* Emite hub:ping cu detail și ascultă pe document.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p05-custom-event-basics/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p05-custom-event-basics/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*06. L1-p06-template-clone — <template> + cloneNode pentru carduri\*\*

\*Specificație.\* Generează carduri folosind <template id='card-tpl'>.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p06-template-clone/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p06-template-clone/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*07. L1-p07-fragment-render — DocumentFragment pentru randare eficientă\*\*

\*Specificație.\* Construiește lista într-un fragment și atașează o singură dată.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p07-fragment-render/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p07-fragment-render/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*08. L1-p08-dataset-attrs — data-\* (dataset) cu validare\*\*

\*Specificație.\* Folosește data-id/data-club pe carduri, validează prezența.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p08-dataset-attrs/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p08-dataset-attrs/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*09. L1-p09-class-toggle — Toggling de clase + ARIA\*\*

\*Specificație.\* La click setează .active și aria-pressed corespunzător.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p09-class-toggle/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p09-class-toggle/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*10. L1-p10-form-input-sync — Form input binding\*\*

\*Specificație.\* Sincronizează input#q cu textul filtrat (fără import lazy).

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p10-form-input-sync/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p10-form-input-sync/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*11. L1-p11-aria-live — ARIA live region\*\*

\*Specificație.\* Anunță numărul de rezultate în aria-live='polite'.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p11-aria-live/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p11-aria-live/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*12. L1-p12-keyboard-events — Keyboard (Enter/Space) pe carduri\*\*

\*Specificație.\* Activează cardul la Enter/Space; focus management minimal.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p12-keyboard-events/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p12-keyboard-events/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*13. L1-p13-target-currentTarget — target vs currentTarget\*\*

\*Specificație.\* Exemplu clar cu event.target vs event.currentTarget.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p13-target-currentTarget/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p13-target-currentTarget/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*14. L1-p14-cleanup-abort — AbortController pentru cleanup\*\*

\*Specificație.\* Returnează dispose() din bootstrap care abort() subscrierile.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p14-cleanup-abort/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p14-cleanup-abort/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*15. L1-p15-module-boundary — Boundary minim între ui/ și services/\*\*

\*Specificație.\* Consumă datele numai prin services/index.js pentru decuplare.

\*Soluție (rezumat).\* Implementarea este în `L1/L1-p15-module-boundary/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L1/L1-p15-module-boundary/tests/`. Consultați `README.md` pentru obiective și rulare.

—

### L2 — Proiecte (15)

\*\*01. L2-p01-import-maps — Import maps (CDN ESM) — demonstrație HTML\*\*

\*Specificație.\* Definește <script type='importmap'> și un exemplu cu lodash-es; testul scanează doar importmap în HTML.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p01-import-maps/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p01-import-maps/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*02. L2-p02-services-json — Strat servicii: loadEvents() + normalizare\*\*

\*Specificație.\* Încarcă /public/data/events.json, normalizează {id,title,when(ISO),club}.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p02-services-json/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p02-services-json/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*03. L2-p03-dynamic-import — Import dinamic pentru filtrare\*\*

\*Specificație.\* Încarcă /src/filters.js la primul input pe #q (lazy).

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p03-dynamic-import/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p03-dynamic-import/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*04. L2-p04-reexports — Re-exports (services/index.js)\*\*

\*Specificație.\* Re-exportă loadEvents; main.js importă exclusiv din index.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p04-reexports/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p04-reexports/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*05. L2-p05-tree-shaking — sideEffects:false (demonstrație teoretică)\*\*

\*Specificație.\* README explică; testele verifică existența câmpului în package.json.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p05-tree-shaking/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p05-tree-shaking/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*06. L2-p06-code-splitting — Buton care încarcă un modul secundar cu import()\*\*

\*Specificație.\* La click pe #feature-load, importă secundary.js și execută.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p06-code-splitting/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p06-code-splitting/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*07. L2-p07-modulepreload-metrics — modulepreload + notițe performanță\*\*

\*Specificație.\* Adaugă <link rel='modulepreload'> și marchează cu performance.now().

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p07-modulepreload-metrics/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p07-modulepreload-metrics/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*08. L2-p08-csp-scan — Igienă CSP: fără on\*=\*\*

\*Specificație.\* Test care scanează HTML pentru on\w+=; README despre CSP.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p08-csp-scan/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p08-csp-scan/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*09. L2-p09-adapter — Adapter/whitelisting pentru JSON\*\*

\*Specificație.\* Mapează doar câmpurile permise, ignoră restul.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p09-adapter/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p09-adapter/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*10. L2-p10-mutation-observer — MutationObserver pentru badge rezultate\*\*

\*Specificație.\* Actualizează un badge când #list se schimbă.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p10-mutation-observer/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p10-mutation-observer/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*11. L2-p11-throttle-input — Throttling pentru input\*\*

\*Specificație.\* Aplică throttle(fn, 80ms) pe inputul de căutare.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p11-throttle-input/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p11-throttle-input/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*12. L2-p12-event-guards — Assert/guards pentru evenimente\*\*

\*Specificație.\* Validează tipul evenimentului înainte de handler.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p12-event-guards/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p12-event-guards/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*13. L2-p13-import-conditions-doc — Import conditions (Node) — explicații\*\*

\*Specificație.\* README + fișier demo (nu rulează în browser).

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p13-import-conditions-doc/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p13-import-conditions-doc/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*14. L2-p14-subpath-exports-doc — Subpath exports — concept\*\*

\*Specificație.\* README + fișiere exemplu; consum în monorepo.

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p14-subpath-exports-doc/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p14-subpath-exports-doc/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*15. L2-p15-esm-cjs-interop-doc — Interop ESM↔CJS — concept\*\*

\*Specificație.\* README + script Node minim (exemplu)

\*Soluție (rezumat).\* Implementarea este în `L2/L2-p15-esm-cjs-interop-doc/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L2/L2-p15-esm-cjs-interop-doc/tests/`. Consultați `README.md` pentru obiective și rulare.

—

### L3 — Proiecte (15)

\*\*01. L3-p01-bus-namespaced — Event bus namespaced (hub:\*)\*\*

\*Specificație.\* Emit/ascultă evenimente pe canale hub:\*; unsubscribe ușor.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p01-bus-namespaced/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p01-bus-namespaced/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*02. L3-p02-timeout-signal — AbortSignal timeout wrapper\*\*

\*Specificație.\* Wrapper care emulează AbortSignal.timeout(ms) dacă nu există.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p02-timeout-signal/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p02-timeout-signal/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*03. L3-p03-intersection-observer — IntersectionObserver pentru lazy UI\*\*

\*Specificație.\* Încarcă detalii pentru carduri abia când devin vizibile.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p03-intersection-observer/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p03-intersection-observer/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*04. L3-p04-mutation-raf-batch — MutationObserver + rAF (batch vizual)\*\*

\*Specificație.\* Batch de actualizări într-un rAF pentru fluiditate.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p04-mutation-raf-batch/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p04-mutation-raf-batch/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*05. L3-p05-composed-events — CustomEvent composed (concept & demo)\*\*

\*Specificație.\* Explică composed:true peste shadow boundary; demo simplu.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p05-composed-events/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p05-composed-events/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*06. L3-p06-event-replay — Event replay buffer\*\*

\*Specificație.\* Păstrează ultimele N evenimente și le livrează noilor abonați.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p06-event-replay/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p06-event-replay/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*07. L3-p07-e2e-filter — Playwright E2E: filtrare listă\*\*

\*Specificație.\* E2E verifică fluxul input→filtrare (webServer).

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p07-e2e-filter/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p07-e2e-filter/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*08. L3-p08-e2e-keyboard — Playwright E2E: navigare tastatură\*\*

\*Specificație.\* E2E verifică Enter/Space/Arrow pe carduri.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p08-e2e-keyboard/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p08-e2e-keyboard/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*09. L3-p09-pwa-workbox — Workbox PWA: precache public/\*\*\*

\*Specificație.\* generateSW cu workbox-build + test smoke pe sw.js.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p09-pwa-workbox/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p09-pwa-workbox/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*10. L3-p10-e2e-offline-smoke — E2E offline smoke (concept)\*\*

\*Specificație.\* Playwright verifică că pagina funcționează după SW.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p10-e2e-offline-smoke/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p10-e2e-offline-smoke/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*11. L3-p11-module-graph — Module graph vizualiser (dev script)\*\*

\*Specificație.\* Traversează importurile din src/ și listează graful.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p11-module-graph/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p11-module-graph/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*12. L3-p12-feature-toggles — Feature toggles cu import() condițional\*\*

\*Specificație.\* Rulează modul opțional doar când toggle activ.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p12-feature-toggles/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p12-feature-toggles/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*13. L3-p13-microfrontend-bridge — iFrame + postMessage ↔ CustomEvent\*\*

\*Specificație.\* Pod între iFrame și document; demo + README.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p13-microfrontend-bridge/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p13-microfrontend-bridge/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*14. L3-p14-perf-budget — Performance budget (marks & asserts)\*\*

\*Specificație.\* Măsoară timpi și eșuează peste prag; doc + test simplu.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p14-perf-budget/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p14-perf-budget/tests/`. Consultați `README.md` pentru obiective și rulare.

—

\*\*15. L3-p15-ci-matrix — CI matrix (doc + workflows monorepo)\*\*

\*Specificație.\* Explică workflows separate și criterii de rulare.

\*Soluție (rezumat).\* Implementarea este în `L3/L3-p15-ci-matrix/src/` și `public/`. Testele unitare (Vitest/Jest) sunt în `L3/L3-p15-ci-matrix/tests/`. Consultați `README.md` pentru obiective și rulare.

—

---

\*\*Rulare (standalone).\*\* În directorul unui proiect: `npm i && npm run test && npm run serve`.

\*\*Rulare (monorepo).\*\* La rădăcina arhivei PNPM: `pnpm i && pnpm -r run test` (unit), iar pentru e2e `pnpm -F @s4/L3-p07-e2e-filter run test:e2e`.

\*\*CI livrat în monorepo.\*\* Workflows separate: `unit-vitest.yml`, `unit-jest.yml`, `e2e.yml`, `workbox-sw.yml`.

\*\*AI‑assist (VSL).\*\* Pentru fiecare proiect, consultați README pentru sugestii de Verify/Specify/Learn.

\*\*Bibliografie (APA 7).\*\* Wirfs‑Brock & Eich (2020, doi:10.1145/3386327); Loring et al. (2017, doi:10.1145/3170472.3133846); Maffeis et al. (2008, doi:10.1007/978-3-540-89330-1\_22); Ungar & Smith (1991, doi:10.1007/BF01806105).