**Ghid de utilizare — Arhive Seminarul 3 (Partea 3)**

Acest ghid descrie conținutul și modul de utilizare pentru cele două arhive livrate în cadrul Seminarului 3 — Partea 3 (Proiecte/teme): varianta Standalone (npm) și varianta Monorepo (PNPM workspaces), precum și workflow‑urile de CI și indicații de depanare.

## 1) Arhiva Standalone — s3-p3-standalone.zip

Conține 45 de proiecte independente organizate pe niveluri: L1, L2, L3. Fiecare proiect include:

• package.json (ESM, scripturi test Vitest/Jest)

• src/index.js (implementarea)

• tests/vitest/spec.test.js și tests/jest/spec.jest.test.js (suite în oglindă)

• README.md (learning goals și pași de rulare)

• .gitignore, babel.config.cjs, jest.config.cjs, vitest.config.ts

Structură orientativă:

s3-p3-standalone/  
 L1/  
 L1-p01-student-vo/  
 package.json  
 src/index.js  
 tests/  
 vitest/spec.test.js  
 jest/spec.jest.test.js  
 README.md  
 L2/  
 L3/ (include proiectele speciale e2e/PWA unde e cazul)

Rulare pentru un proiect:

npm i  
npm run test

Pentru proiectele E2E (L3‑p07/p14/p15):

npm run serve # într-un terminal (pornește serverul)  
npm run test:e2e # în alt terminal

## 2) Arhiva Monorepo (PNPM workspaces) — s3-p3-monorepo.zip

Workspaces în directorul „packages/”. Pachetul rădăcină conține dependențe comune și workflow‑uri CI.

Structură orientativă:

s3-p3-monorepo/  
 package.json  
 pnpm-workspace.yaml  
 .github/workflows/  
 unit-vitest.yml  
 unit-jest.yml  
 e2e.yml  
 workbox-sw.yml  
 packages/  
 L1-p01-student-vo/  
 package.json  
 src/index.js  
 tests/...  
 L3-p07-e2e-site/  
 package.json  
 server.js  
 public/index.html  
 tests/e2e.spec.ts  
 playwright.config.ts  
 L3-p08-pwa-workbox/  
 package.json  
 public/index.html  
 build-sw.mjs  
 tests/pwa.spec.ts  
 playwright.config.ts

Instalare și rulare la rădăcină:

pnpm i  
pnpm run test # rulează Vitest + Jest pe toate pachetele

Comenzi utile pentru pachetele speciale:

pnpm -F @s3/L3-p07-e2e-site exec playwright test  
pnpm -F @s3/L3-p08-pwa-workbox exec node build-sw.mjs

## 3) Workflow‑uri CI (GitHub Actions)

În `.github/workflows/` sunt definite joburi separate pentru tipuri de teste diferite:

• unit-vitest.yml — rulează suita Vitest pentru toate pachetele

• unit-jest.yml — rulează suita Jest în oglindă

• e2e.yml — instalează Playwright și execută testele E2E pentru pachetele L3 (site, table, form)

• workbox-sw.yml — generează `public/sw.js` în pachetul PWA și verifică existența fișierului

Note: în E2E se folosește un `webServer` (sau `serve` script); în CI, browserele Playwright se instalează cu `npx playwright install --with-deps`.

## 4) Troubleshooting

• Dacă Jest raportează probleme cu ESM: verifică `extensionsToTreatAsEsm` și transformatorul `babel-jest`.

• Dacă testele pe timp (`setTimeout`) sunt instabile pe mașini lente/CI: crește latențele din teste (ex. 5→15ms).

• Pentru E2E: asigură-te că serverul este pornit (`npm run serve`) sau configurat ca `webServer` în `playwright.config.ts`.

• Pentru PWA/Workbox: rulează `node build-sw.mjs` înainte de testele care verifică `public/sw.js`.

• Pe Windows: rulează terminalul în mod cu permisiuni suficiente și folosește `pnpm i --shamefully-hoist` dacă întâmpini probleme de rezoluție.

Întrebări? Verifică README‑urile fiecărui proiect și fișierele de configurare din arhive.