**Seminarul 5 — CSS Selectors & Layout  
Partea 2: Laborator (extins)**

**Context, obiective și criterii de evaluare**

Această parte a laboratorului transformă conceptele teoretice din Partea 1 într‑o practică aplicată, folosind o temă comună: „Clubs & Associations Hub”. Vei construi o pagină cu un \*\*grid 3×3\*\* de carduri și un \*\*formular de înscriere\*\* a studenților, de la simplu la complex, exclusiv cu \*\*CSS selectors\*\* bine alese, \*\*pseudo‑clase\*\* orientate pe stări de UI și \*\*layout\*\* modern cu \*\*Grid\*\*. Pe parcurs, menținem un nivel înalt de lizibilitate și accesibilitate (focus vizibil, contraste, spațiere), susținute de principii HCI.

\*\*Obiectivul\*\* este să practici: (1) alegerea selectorilor cu \*\*specificitate mică\*\*, (2) folosirea pseudo‑claselor moderne (`:is()`, `:where()`, `:has()`, `:focus-visible`, `:focus-within`, `:invalid/:valid`, `:placeholder-shown`), (3) compunerea unui \*\*grid 3×3\*\* responsiv, și (4) un \*\*form styling\*\* robust care valorifică stările native ale browserului.

\*\*Criterii de evaluare\*\* (rezumat): (a) corectitudinea vizuală și funcțională (focus ring, stări de formular, grid 3×3), (b) simplitatea și calitatea selectorilor (fără `!important`, lanțuri scurte, `:where()` când e cazul), (c) accesibilitate (contrast, mărimea „targets”, `aria-live` pentru erori), (d) întreținere (BEM, tokens).

**Prerechizite & setup local**

Ai nevoie de \*\*Node.js 20+\*\*, un manager de pachete (npm sau pnpm), și un editor care suportă GitHub Copilot/LLM. Pentru a porni instant, folosește \*\*arhiva starter\*\* livrată cu această parte sau creează structura din secțiunea „Starter code”. Rulează serverul minimal (Express) pentru a servi fișierele statice și pentru a avea un URL local stabil (necesar testelor e2e în alte seminarii).

npm i  
npm run dev # http://localhost:5175 (redirect la /public/index.html)

**Arhitectura folderelor (starter)**

Arborele recomandat delimitează clar resursele vizuale (CSS), structura (HTML) și interactivitatea minimă (JS). Testele sunt organizate în `tests/unit` și rulează atât pe \*\*Vitest\*\*, cât și pe \*\*Jest\*\* (jsdom).

s5p2-starter/  
 public/index.html  
 styles/{base.css, forms.css, layout.css, components.css}  
 src/{utils/dom.js, main.js}  
 tests/unit/{selectors.vitest.test.ts, selectors.jest.test.mjs}  
 vitest.config.ts  
 jest.config.cjs  
 server.mjs  
 package.json

**Worksheet — enunț + checklist (de predat)**

\*\*Enunț\*\*: Implementați pagina „Clubs & Associations Hub” folosind exclusiv \*\*CSS selectors\*\* și \*\*CSS Grid\*\* (JS doar pentru feedback „live” pe formular). Păstrați specificitatea scăzută și evitați `!important`. Formularul trebuie să evidențieze local câmpurile invalide, să aibă focus ring vizibil și un sumar textual al erorilor. Gridul cardurilor va fi 3×3 pe desktop, colapsând la 2×? sub 900px și la 1 coloană sub 520px, cu imagini menținând raportul 4:3.

\*\*Checklist\*\* (bifat la predare):

• Există `:focus-visible` pe linkuri/butoane/inputuri.  
• Grupurile de formular folosesc `:has(input:invalid)` pentru highlight.  
• Form conține `novalidate`, câmpuri `required` și mesaje de eroare locale (`role="status"`/`aria-live`).  
• Grid \*\*3×3\*\* este definit prin `display: grid` + `grid-template-columns: repeat(3, 1fr)` și colapsează corect.  
• S-au utilizat `:is()`/`:where()` pentru a limita specificitatea lanțurilor lungi.  
• Nu s-a folosit `!important`.

**Starter code (HTML + CSS + JS)**

Fișierele de mai jos reprezintă \*\*varianta de start\*\*. Poți copia exact sau poți porni de la ele și itera. \*\*Observație\*\*: toate fragmentele de cod din acest document se regăsesc în arhiva starter generată.

`public/index.html` — structură semantică + grid de carduri + formular de înscriere.

<!doctype html>  
<html lang="ro">  
 <head>  
 <meta charset="utf-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1" />  
 <title>Seminar 5 — CSS Lab (Selectors, Forms, Grid 3×3)</title>  
 <link rel="stylesheet" href="../styles/base.css" />  
 <link rel="stylesheet" href="../styles/forms.css" />  
 <link rel="stylesheet" href="../styles/layout.css" />  
 <link rel="stylesheet" href="../styles/components.css" />  
 </head>  
 <body>  
 <header class="site-header">  
 <h1>Clubs &amp; Associations Hub</h1>  
 <p class="lead">Descoperă cluburile, înscrie-te rapid, rămâi conectat.</p>  
 <nav aria-label="Sarituri rapide">  
 <a class="skip" href="#join">Sari la formular</a>  
 <a class="skip" href="#cards">Sari la carduri</a>  
 </nav>  
 </header>  
  
 <main id="app" class="container">  
 <section id="cards" aria-labelledby="cards-title">  
 <h2 id="cards-title">Cluburi active</h2>  
 <ul class="cards">  
 <!-- 9 carduri pentru grid 3×3 -->  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?1" alt="Club 1" /></div>  
 <h3 class="card\_\_title">Club 1</h3>  
 <p class="card\_\_desc">Debate &amp; Rhetoric.</p>  
 <span class="card\_\_badge" aria-label="inscrieri deschise"></span>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?2" alt="Club 2" /></div>  
 <h3 class="card\_\_title">Club 2</h3>  
 <p class="card\_\_desc">Tech &amp; Coding.</p>  
 <span class="card\_\_badge" aria-label="inscrieri deschise"></span>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?3" alt="Club 3" /></div>  
 <h3 class="card\_\_title">Club 3</h3>  
 <p class="card\_\_desc">Arts &amp; Culture.</p>  
 <span class="card\_\_badge" aria-label="inscrieri deschise"></span>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?4" alt="Club 4" /></div>  
 <h3 class="card\_\_title">Club 4</h3>  
 <p class="card\_\_desc">Volunteering.</p>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?5" alt="Club 5" /></div>  
 <h3 class="card\_\_title">Club 5</h3>  
 <p class="card\_\_desc">Music &amp; Band.</p>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?6" alt="Club 6" /></div>  
 <h3 class="card\_\_title">Club 6</h3>  
 <p class="card\_\_desc">Entrepreneurship.</p>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?7" alt="Club 7" /></div>  
 <h3 class="card\_\_title">Club 7</h3>  
 <p class="card\_\_desc">Sports &amp; Health.</p>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?8" alt="Club 8" /></div>  
 <h3 class="card\_\_title">Club 8</h3>  
 <p class="card\_\_desc">Photography.</p>  
 </li>  
 <li class="card">  
 <div class="card\_\_media"><img src="https://picsum.photos/400/300?9" alt="Club 9" /></div>  
 <h3 class="card\_\_title">Club 9</h3>  
 <p class="card\_\_desc">Eco &amp; Sustainability.</p>  
 </li>  
 </ul>  
 </section>  
  
 <section id="join" aria-labelledby="join-title" class="join">  
 <h2 id="join-title">Formular de înscriere</h2>  
 <form id="join-form" novalidate>  
 <div class="form-grid">  
 <div class="form-group field">  
 <label for="name">Nume complet <span aria-hidden="true">\*</span></label>  
 <input id="name" name="name" required placeholder="ex.: Popescu Andrei" />  
 <small class="hint">Introduceți numele complet.</small>  
 <small class="error" id="err-name" role="status" aria-live="polite"></small>  
 </div>  
  
 <div class="form-group field">  
 <label for="email">Email instituțional <span aria-hidden="true">\*</span></label>  
 <input id="email" type="email" name="email" required placeholder="ex.: nume@stud.ase.ro" />  
 <small class="hint">Folositi adresa instituțională.</small>  
 <small class="error" id="err-email" role="status" aria-live="polite"></small>  
 </div>  
  
 <div class="form-group field">  
 <label for="phone">Telefon</label>  
 <input id="phone" name="phone" inputmode="numeric" pattern="^[0-9+()\-\s]{6,}$" placeholder="ex.: 07xx ..." />  
 <small class="hint">Opțional; minim 6 caractere numerice.</small>  
 <small class="error" id="err-phone" role="status" aria-live="polite"></small>  
 </div>  
  
 <div class="form-group field">  
 <label for="club">Club preferat <span aria-hidden="true">\*</span></label>  
 <select id="club" name="club" required>  
 <option value="">Selectează...</option>  
 <option>Debate &amp; Rhetoric</option>  
 <option>Tech &amp; Coding</option>  
 <option>Arts &amp; Culture</option>  
 <option>Volunteering</option>  
 <option>Music &amp; Band</option>  
 <option>Entrepreneurship</option>  
 <option>Sports &amp; Health</option>  
 <option>Photography</option>  
 <option>Eco &amp; Sustainability</option>  
 </select>  
 <small class="error" id="err-club" role="status" aria-live="polite"></small>  
 </div>  
  
 <div class="form-group field">  
 <label for="file">CV (PDF, opțional)</label>  
 <input id="file" type="file" accept="application/pdf" />  
 <small class="hint">Maxim 2 MB (verificare manuală).</small>  
 </div>  
  
 <div class="form-group field">  
 <label>  
 <input id="consent" type="checkbox" required />  
 Sunt de acord cu prelucrarea datelor (GDPR)  
 </label>  
 <small class="error" id="err-consent" role="status" aria-live="polite"></small>  
 </div>  
 </div>  
  
 <div class="form-actions">  
 <button type="submit" class="btn">Trimite</button>  
 <button type="reset" class="btn btn--secondary">Resetează</button>  
 </div>  
  
 <div class="errors" role="status" aria-live="polite"></div>  
 </form>  
 </section>  
 </main>  
  
 <footer class="site-footer">  
 <small>&copy; 2025 TW • Seminar 5</small>  
 </footer>  
  
 <script type="module" src="../src/main.js"></script>  
 </body>  
</html>

`styles/base.css` — tokens, tipografie, focus ring, container.

:root{  
 color-scheme: light dark;  
 --bg: #ffffff;  
 --fg: #111827;  
 --muted: #6b7280;  
 --primary: #1a73e8;  
 --success: #1e8e3e;  
 --error: #b00020;  
 --card: #ffffff;  
 --border: #e5e7eb;  
 --space-2: .5rem;  
 --space-4: 1rem;  
 --space-6: 1.5rem;  
 --radius: .75rem;  
 --ring: 2px solid var(--primary);  
}  
@media (prefers-color-scheme: dark){  
 :root{  
 --bg: #0b0f14;  
 --fg: #e5e7eb;  
 --card: #10151b;  
 --border: #263241;  
 }  
}  
\*{ box-sizing: border-box; }  
html, body{ margin:0; padding:0; }  
body{ font-family: system-ui, -apple-system, Segoe UI, Roboto, sans-serif; line-height:1.5; color:var(--fg); background:var(--bg); }  
h1,h2,h3{ line-height:1.2; }  
a{ color: var(--primary); text-decoration: none; }  
a:hover, a:focus-visible{ text-decoration: underline; }  
  
/\* Focus ring consecvent pe elemente interactive \*/  
:where(a, button, input, select, textarea):focus-visible{  
 outline: var(--ring);  
 outline-offset: 2px;  
}  
  
.container{ max-width: 72rem; margin: 0 auto; padding: var(--space-6) var(--space-4); }  
.site-header{ padding: var(--space-6) var(--space-4) var(--space-4); }  
.site-header .lead{ color: var(--muted); }  
.skip{ margin-right: .75rem; font-size: .875rem; }  
  
.site-footer{ padding: var(--space-6) var(--space-4); color: var(--muted); }  
.visually-hidden{ position:absolute !important; width:1px; height:1px; overflow:hidden; clip:rect(1px,1px,1px,1px); clip-path: inset(50%); white-space:nowrap; }

`styles/forms.css` — stări native (`:invalid/:valid`), `:focus-within`, `:has()` pentru grupuri, `::file-selector-button`, `accent-color`.

/\* Form styling \*/  
:where(input, select, textarea){  
 width: 100%;  
 padding: .625rem .75rem;  
 border: 1px solid var(--border);  
 border-radius: .5rem;  
 background: var(--card);  
 color: var(--fg);  
}  
label{ display:block; font-weight: 600; margin-bottom: .375rem; }  
.hint{ color: var(--muted); }  
.error{ color: var(--error); min-height: 1.25rem; display:block; }  
  
/\* Accent pentru checkbox/radio \*/  
input[type="checkbox"], input[type="radio"]{ accent-color: var(--primary); width:auto; }  
  
/\* File input button \*/  
input[type="file"]::file-selector-button{  
 padding: .5rem .75rem;  
 border: 1px solid var(--border);  
 background: #f3f4f6;  
 color: #111827;  
 border-radius: .5rem;  
 cursor: pointer;  
}  
  
/\* Stări nativ validate/invalid \*/  
:where(input, select, textarea):required:invalid{ border-color: var(--error); }  
:where(input, select, textarea):required:valid{ border-color: var(--success); }  
  
/\* :focus-within la nivel de grup \*/  
.field:has(:focus){ box-shadow: 0 0 0 2px color-mix(in oklab, var(--primary) 20%, transparent); border-radius: .5rem; }  
  
/\* :has() pentru grup cu erori \*/  
.form-group:has(input:invalid), .form-group:has(select:invalid){  
 background: color-mix(in oklab, var(--error) 6%, transparent);  
 padding: .25rem;  
 border-radius: .5rem;  
}  
  
/\* Placeholder vizibil doar când e gol câmpul \*/  
:where(input, textarea):placeholder-shown{ color: #9ca3af; }  
  
.form-grid{  
 display: grid;  
 grid-template-columns: repeat(2, minmax(0, 1fr));  
 gap: var(--space-4);  
}  
@media (max-width: 640px){  
 .form-grid{ grid-template-columns: 1fr; }  
}  
  
.form-actions{  
 position: sticky;  
 bottom: 0;  
 background: color-mix(in oklab, var(--bg) 85%, transparent);  
 display: flex; gap: .75rem; padding: .5rem; margin-top: var(--space-4);  
 border-top: 1px solid var(--border);  
}  
.errors{ margin-top: .5rem; min-height: 1.25rem; }

`styles/layout.css` — grid 3×3 + responsive collapse, carduri cu `:hover/:focus-within`.

.cards{  
 display: grid;  
 grid-template-columns: repeat(3, 1fr);  
 gap: var(--space-6);  
}  
@media (max-width: 900px){ .cards{ grid-template-columns: repeat(2, 1fr); } }  
@media (max-width: 520px){ .cards{ grid-template-columns: 1fr; } }  
  
.card{  
 background: var(--card);  
 border: 1px solid var(--border);  
 border-radius: var(--radius);  
 padding: var(--space-4);  
 display: grid;  
 gap: var(--space-2);  
}  
.card\_\_media{ aspect-ratio: 4/3; overflow:hidden; border-radius: .5rem; }  
.card\_\_media > img{ width:100%; height:100%; object-fit: cover; display:block; }  
.card\_\_title{ margin:0; font-size: 1.125rem; }  
.card\_\_desc{ margin:0; color: var(--muted); }  
.card:where(:hover, :focus-within){ border-color: var(--primary); box-shadow: 0 2px 10px rgba(0,0,0,.08); }  
.card\_\_badge::before{  
 content: "Open";  
 font-size: .75rem;  
 display: inline-block; padding: .125rem .5rem;  
 background: var(--success); color: white;  
 border-radius: 999px;  
}

`styles/components.css` — butoane și shell.

.btn{  
 --bg: var(--primary);  
 --fg: #fff;  
 appearance: none;  
 border: 1px solid transparent;  
 background: var(--bg);  
 color: var(--fg);  
 padding: .625rem 1rem;  
 border-radius: .5rem;  
 cursor: pointer;  
}  
.btn--secondary{ --bg: #4b5563; }  
.btn:hover, .btn:focus-visible{ filter: brightness(1.05); }  
  
.site-header, .site-footer{ border-top: 1px solid var(--border); border-bottom: 1px solid var(--border); }

`src/utils/dom.js` — utilitare simple pentru DOM (qs/qsa/on/delegate).

export const qs = (sel, root=document) => root.querySelector(sel);  
export const qsa = (sel, root=document) => [...root.querySelectorAll(sel)];  
export function on(target, type, handler, opts={}){  
 const { signal, capture=false, once=false, passive=false } = opts;  
 target.addEventListener(type, handler, { capture, once, passive, signal });  
 return () => target.removeEventListener(type, handler, { capture });  
}  
export function delegate(root, type, selector, handler, opts={}){  
 const proxy = (e) => {  
 const el = e.target.closest(selector);  
 if(el && root.contains(el)) handler(e, el);  
 };  
 return on(root, type, proxy, opts);  
}

`src/main.js` — validare „live” minimală și mesaj sumar de erori.

import { qs, qsa, on } from './utils/dom.js';  
  
const form = qs('#join-form');  
const errorsRegion = qs('.errors', form);  
  
function validateField(el){  
 const errorEl = el.id ? form.querySelector(`#err-${el.id}`) : null;  
 if(!el.checkValidity()){  
 if(errorEl){ errorEl.textContent = el.validationMessage; }  
 return false;  
 } else {  
 if(errorEl){ errorEl.textContent = ''; }  
 return true;  
 }  
}  
  
if(form){  
 // live validation on blur / input (lightweight)  
 qsa('input, select, textarea', form).forEach(el => {  
 on(el, 'blur', () => validateField(el));  
 on(el, 'input', () => { if(!el.matches(':placeholder-shown')) validateField(el); });  
 });  
  
 on(form, 'submit', (e) => {  
 e.preventDefault();  
 const fields = qsa('input, select, textarea', form);  
 const invalid = fields.filter(f => !validateField(f));  
 if(invalid.length){  
 errorsRegion.textContent = `Verificați câmpurile marcate (${invalid.length}).`;  
 invalid[0].focus();  
 return;  
 }  
 errorsRegion.textContent = 'Trimis! Vă contactăm curând.';  
 form.reset();  
 });  
}

**Exerciții progresive (L1 → L2 → L3) — pași detaliați**

\*\*L1 — Bazele selectors + stări vizibile\*\*  
1) Activează `:focus-visible` pentru linkuri, butoane și inputuri (în `base.css`). Verifică vizibilitatea la navigare cu tastatura (Tab/Shift+Tab).   
2) În `layout.css`, setează gridul 3×3 pe `.cards` și aplică un `gap` confortabil. Adaugă `:hover/:focus-within` pe `.card` pentru feedback.   
3) În `forms.css`, asigură-te că `input:required:invalid` și `input:required:valid` schimbă `border-color`.   
4) Rulează testele; dacă unul eșuează, citește mesajul și corectează.

\*\*L2 — Compoziție modernă (`:is()`, `:where()`) + parent styling (`:has()`)\*\*  
5) Înlocuiește lanțuri adânci de selectori cu `:where(...)` în prefață pentru a reduce specificitatea (ex.: `:where(.card) > h3 + p`).   
6) Adaugă `:focus-within` pe `.field` și `:has(input:invalid)` pe `.form-group` pentru evidențierea locală a erorilor.   
7) Stilizează `input[type="file"]::file-selector-button` și setează `accent-color` pentru `checkbox/radio`.   
8) Ajustează `form-actions` la `position: sticky` pentru a „ancora” acțiunile în viewport.   
9) Re‑rulează testele unitare.

\*\*L3 — Responsivitate & fine‑tuning UX\*\*  
10) Adaugă media queries pentru colapsul 3→2→1; verifică manual la 900px și 520px.   
11) Adaugă `aspect-ratio` pe `.card\_\_media` și `object-fit: cover` pe `<img>`.   
12) Folosește tokens (`--space-\*`, `--radius`, `--primary`) pentru a crea rapid o temă ușoară/dark (prin `prefers-color-scheme`).   
13) Testează pe mobil (DevTools) focus ring (Tab order/keyboard nav) și „target size” (buton/checkbox).

**Unit tests (Vitest & Jest, side‑by‑side) — verificări de convenții**

Scopul testelor este \*\*să valideze convenții\*\* și prezența selectorilor/stărilor în fișierele CSS/HTML, nu să „calculeze” stilurile efective (acest lucru aparține testării e2e). Testele rulează cu \*\*jsdom\*\* și verifică: (a) `forms.css` conține `:focus-visible`, `:has()`, `:invalid`; (b) `layout.css` definește gridul 3×3; (c) `index.html` are formular `novalidate` și câmpuri `required`.

`vitest.config.ts`

import { defineConfig } from 'vitest/config';  
export default defineConfig({  
 test: {  
 environment: 'jsdom'  
 }  
});

`jest.config.cjs`

module.exports = {  
 testEnvironment: 'jsdom'  
};

`tests/unit/selectors.vitest.test.ts`

import { describe, it, expect } from 'vitest';  
import fs from 'node:fs';  
import path from 'node:path';  
import { JSDOM } from 'jsdom';  
  
const root = path.resolve(\_\_dirname, '..', '..');  
  
describe('CSS selectors & layout — presence checks', () => {  
 it('forms.css include :focus-visible, :has(), :invalid', () => {  
 const css = fs.readFileSync(path.join(root, 'styles', 'forms.css'), 'utf8');  
 expect(css).toMatch(/:focus-visible/);  
 expect(css).toMatch(/:has\(/);  
 expect(css).toMatch(/:invalid/);  
 });  
  
 it('layout.css include grid 3×3', () => {  
 const css = fs.readFileSync(path.join(root, 'styles', 'layout.css'), 'utf8');  
 expect(css).toMatch(/display:\s\*grid/);  
 expect(css).toMatch(/grid-template-columns:\s\*repeat\(3,\s\*1fr\)/);  
 });  
  
 it('index.html are un formular cu campuri required', () => {  
 const html = fs.readFileSync(path.join(root, 'public', 'index.html'), 'utf8');  
 const dom = new JSDOM(html);  
 const doc = dom.window.document;  
 expect(doc.querySelector('form#join-form[novalidate]')).toBeTruthy();  
 expect(doc.querySelectorAll('#join-form [required]').length).toBeGreaterThanOrEqual(3);  
 });  
});

`tests/unit/selectors.jest.test.mjs`

const fs = require('node:fs');  
const path = require('node:path');  
const { JSDOM } = require('jsdom');  
  
const root = path.resolve(\_\_dirname, '..', '..');  
  
test('forms.css include :focus-visible, :has(), :invalid', () => {  
 const css = fs.readFileSync(path.join(root, 'styles', 'forms.css'), 'utf8');  
 expect(css).toMatch(/:focus-visible/);  
 expect(css).toMatch(/:has\(/);  
 expect(css).toMatch(/:invalid/);  
});  
  
test('layout.css include grid 3×3', () => {  
 const css = fs.readFileSync(path.join(root, 'styles', 'layout.css'), 'utf8');  
 expect(css).toMatch(/display:\s\*grid/);  
 expect(css).toMatch(/grid-template-columns:\s\*repeat\(3,\s\*1fr\)/);  
});  
  
test('index.html are formular cu required', () => {  
 const html = fs.readFileSync(path.join(root, 'public', 'index.html'), 'utf8');  
 const dom = new JSDOM(html);  
 const doc = dom.window.document;  
 expect(doc.querySelector('form#join-form[novalidate]')).toBeTruthy();  
 expect(doc.querySelectorAll('#join-form [required]').length).toBeGreaterThanOrEqual(3);  
});

**Scripturi NPM și rulare testelor**

Folosește scripturile din `package.json` pentru a rula local:

npm run dev  
npm run test:vitest  
npm run test:jest  
npm test

**Depanare (troubleshooting) — cazuri frecvente**

• \*\*Focus ring „dispare”\*\*: folosești `:focus` în loc de `:focus-visible`? Asigură-te că nu l-ai anulat cu `outline: none`.   
• \*\*Grid nu colapsează corect\*\*: verifică media queries, dar și dacă CSS este încărcat în ordinea dorită; `@media` mai specifice pot suprascrie.   
• \*\*`:has()` nu pare să funcționeze\*\*: confirmă suportul în browserul de test; în testele unitare noi verificăm doar \*\*prezența în CSS\*\*. Pentru compat, poți adăuga o clasă suplimentară `is-invalid` din JS ca fallback (nu obligatoriu în acest lab).   
• \*\*Specificitate neașteptată\*\*: mută prefixul de lanț într-un `:where(...)`.   
• \*\*Contrast slab\*\*: ajustează tokens (`--fg`, `--bg`, `--primary`) până atingi ~4.5:1 pentru text normal.

**AI‑assist (Copilot/LLM) — promturi VSL**

• „Propune un set minimal de reguli pentru focus vizibil pe linkuri/butoane/inputuri, folosind `:where()` astfel încât specificitatea totală să fie minimă.”   
• „Generează `display: grid` cu 3 coloane de lățime egală și `gap: 1.5rem`, plus media queries pentru 900px și 520px.”   
• „Rescrie acești trei selectori nestinguiți adânc într-o variantă cu maxim doi combinatori și `:is()`/`:where()`.”   
• „Scrie un bloc de CSS pentru `:has(input:invalid)` pe `.form-group` care schimbă fundalul subtil și border-ul.”

**Rubrică (scor orientativ)**

• \*\*Funcționalitate & UX\*\* (40%): focus vizibil, erori locale, grid 3×3 responsiv.   
• \*\*Calitatea CSS\*\* (35%): specificitate mică, fără `!important`, `:is()`/`:where()`/`:has()` folosite judicios, BEM & tokens.   
• \*\*Accesibilitate\*\* (15%): contrast, target size, `aria-live` pentru erori.   
• \*\*Testare & Disciplina de livrare\*\* (10%): trecerea testelor unitare, structură curată.