**Seminarul 5 — Partea 3: Proiecte/teme (extins)**

**Introducere și metodologie**

Această parte livrează \*\*45 de proiecte\*\* (15 × L1, 15 × L2, 15 × L3) concepute pentru a consolida „gândirea în selectori” și fluența în layout‑uri moderne. Fiecare proiect are o \*\*temă unică\*\*, dar toate gravitează în jurul aceleiași aplicații narative — „Clubs & Associations Hub”: listări în \*\*grid 3×3\*\*, formulare accesibile, componente CSS-only și theming cu \*\*CSS custom properties\*\*. Pentru fiecare proiect oferim: \*\*Scop didactic\*\*, \*\*Specificații (Cerințe)\*\*, \*\*Criterii de acceptare\*\*, \*\*Soluție (rezumat)\*\* și \*\*AI‑assist prompts\*\* (VSL — very short loop).

În fișierele livrate găsiți atât o \*\*arhivă „standalone”\*\* (câte un mini‑repo per proiect), cât și o \*\*variantă monorepo (PNPM workspaces)\*\*. Toate proiectele conțin \*\*teste unitare\*\* (Vitest + Jest, jsdom) orientate pe \*\*convenții\*\*: verifică existența selectorilor și pattern‑urilor relevante în `styles.css` și a structurilor necesare în `public/index.html`. Fiecare proiect are `tests/config.json` — un mic profil al așteptărilor, pe care testerul îl citește pentru a valida \*\*exact\*\* acele pattern‑uri.

**Proiecte L1 (Fundamente) — 15 teme**

**L1 — P01. Grid 3×3 de carduri — structură de bază**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: display:\s\*grid, grid-template-columns:\s\*repeat\(3,\s\*1fr\). HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Definește `.cards { display: grid; grid-template-columns: repeat(3, 1fr); gap: 1rem; }` și media queries pentru 3→2→1.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P02. Focus vizibil universal — linkuri, butoane, inputuri**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: :focus-visible. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Uniformizează `:focus-visible` pentru linkuri/butoane/inputuri cu `outline` și `outline-offset`.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P03. Form minimal — :required/:optional și :placeholder-shown**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :invalid, :valid, :focus-within, :placeholder-shown. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Folosește `:required/:valid/:invalid`, `:placeholder-shown` și `:focus-within` pentru feedback local.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P04. Badge-uri pe atribute — [data-status]**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: \[data-, \[\w+[\^\$\\*]?=. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P05. Liste „zebra” — :nth-child(odd/even)**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: :nth-. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P06. Link states — :visited, :hover, :focus-visible**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: :focus-visible. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Uniformizează `:focus-visible` pentru linkuri/butoane/inputuri cu `outline` și `outline-offset`.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P07. Button states — :hover/:active**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P08. Checkbox/Radio — accent-color + fieldset/legend**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: accent-color. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Setează `accent-color` și etichetează clar opțiunile.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P09. File input — ::file-selector-button (basic)**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: ::file-selector-button. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Stilizează `::file-selector-button` și păstrează contrastul suficient.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P10. Grupuri cu :focus-within (form groups)**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: :focus-visible. CSS include \*\*cel puțin unu\*\* dintre: :invalid, :valid, :focus-within, :placeholder-shown. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Uniformizează `:focus-visible` pentru linkuri/butoane/inputuri cu `outline` și `outline-offset`. Folosește `:required/:valid/:invalid`, `:placeholder-shown` și `:focus-within` pentru feedback local.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P11. Erori locale — input:invalid + mesaje**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P12. Selecții negative — :not() pentru simplificare**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P13. Secțiuni ancoră — :target highlight**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P14. BEM naming pentru componente**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L1 — P15. Tokens (CSS custom properties) + light/dark**

\*\*Scop didactic.\*\* Formarea reflexelor pentru selectori simpli și layout predictibil.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**Proiecte L2 (Intermediare) — 15 teme**

**L2 — P01. :is() și :where() — scăderea specificității**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :is\(, :where\(. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Rescrie lanțurile adânci cu `:where()` pentru a neutraliza specificitatea și `:is()` pentru a grupa stările.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P02. Parent highlight cu :has(input:invalid)**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: :has\(. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică `:has(input:invalid)` pe `.form-group` pentru a colora containerul doar la erori.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P03. Form actions — position: sticky**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :invalid, :valid, :focus-within, :placeholder-shown. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Folosește `:required/:valid/:invalid`, `:placeholder-shown` și `:focus-within` pentru feedback local.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P04. Grid 3×3 responsiv — 3→2→1**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: display:\s\*grid, grid-template-columns:\s\*repeat\(3,\s\*1fr\), @media, grid-template-columns. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Definește `.cards { display: grid; grid-template-columns: repeat(3, 1fr); gap: 1rem; }` și media queries pentru 3→2→1.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P05. Card media — aspect-ratio + object-fit**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: aspect-ratio. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Impune `aspect-ratio` pe media pentru rapoarte consecvente.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P06. Grid areas — layout semantic**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: grid-template-areas. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Definește `.cards { display: grid; grid-template-columns: repeat(3, 1fr); gap: 1rem; }` și media queries pentru 3→2→1. Mapează layoutul cu `grid-template-areas` pentru auto‑documentare.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P07. Range slider — thumb & track**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: ::-webkit-slider-thumb, ::-moz-range-thumb. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P08. Rating control (radio) + accent-color**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: accent-color. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Setează `accent-color` și etichetează clar opțiunile.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P09. File input custom (avansat)**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: ::file-selector-button. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Stilizează `::file-selector-button` și păstrează contrastul suficient.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P10. Tabs doar CSS — :target / :checked**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P11. Accordion doar CSS — :checked + sibling**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P12. Tabel comparativ — :nth-of-type**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: :nth-. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P13. Attribute selectors avansate — ^=, $=, \*=, i**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: \[data-, \[\w+[\^\$\\*]?=. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P14. Dark mode preferențial — prefers-color-scheme**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L2 — P15. Print stylesheet — versiune tipărire**

\*\*Scop didactic.\*\* Compoziții cu selectori moderni, ergonomie și responsive patterns.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: @media\s+print. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**Proiecte L3 (Avansate) — 15 teme**

**L3 — P01. Design tokens pack — tematici light/dark**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P02. Container queries (inline-size) — layout adaptiv**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: container-type:\s\*inline-size. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Activează `container-type: inline-size` și scrie o regulă `@container` pentru adaptare fină.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P03. Grid areas complex — pagina „Club details”**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: grid-template-areas. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Definește `.cards { display: grid; grid-template-columns: repeat(3, 1fr); gap: 1rem; }` și media queries pentru 3→2→1. Mapează layoutul cu `grid-template-areas` pentru auto‑documentare.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P04. Nested grids — card compus**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Definește `.cards { display: grid; grid-template-columns: repeat(3, 1fr); gap: 1rem; }` și media queries pentru 3→2→1.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P05. Form states matrix — combinarea stărilor**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :invalid, :valid, :focus-within, :placeholder-shown. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Folosește `:required/:valid/:invalid`, `:placeholder-shown` și `:focus-within` pentru feedback local.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P06. High contrast & reduced motion**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P07. Density toggle — compact vs. cozy**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P08. Listă tip „email” — :has(:checked) pe rând**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: :has\(. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică `:has(input:invalid)` pe `.form-group` pentru a colora containerul doar la erori.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P09. Scroll snap — carousel de carduri**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: scroll-snap-type. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P10. Pseudo‑elemente pentru iconografie**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P11. Print avansat — page breaks, orphans/widows**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include toate pattern‑urile: @media\s+print. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P12. Masonry‑like grid — auto-rows hack**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Definește `.cards { display: grid; grid-template-columns: repeat(3, 1fr); gap: 1rem; }` și media queries pentru 3→2→1.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P13. Overflow & truncation — ellipsis, hyphenation**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: text-overflow:\s\*ellipsis, overflow:\s\*hidden. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P14. Theming extins — palete și aliasuri**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**L3 — P15. Layout stress test — reziliență la conținut**

\*\*Scop didactic.\*\* Arhitecturi CSS scalabile: theming, queries, scenarii avansate.

\*\*Specificații (Cerințe).\*\* Creează pagina proiectului în baza skeleton‑ului (`public/index.html`, `styles.css`). Menține specificitatea scăzută; evită `!important`. Respectă tema proiectului (selectori/Layout) și \*\*criteriile de acceptare\*\* de mai jos.

\*\*Criterii de acceptare.\*\* CSS include \*\*cel puțin unu\*\* dintre: :focus-visible, display:\s\*grid. HTML include: <main id="app".

\*\*Soluție (rezumat).\*\* Aplică selectori moderni și layout‑ul potrivit temei, păstrând specificitatea mică.

\*\*AI‑assist (VSL prompts).\*\* Exemplu: „Rescrie selectorii de stări astfel încât să folosesc `:where()` + `:is()` și să păstrez specificitatea minimă”; „Propune media queries pentru colaps 3→2→1 cu lățimi 900px/520px”; „Sugerează reguli `:has(input:invalid)` elegante pentru evidențiere locală a erorilor.”

**Rubrică (evaluare)**

Funcționalitate & UX (40%), Calitatea CSS (35%), Accesibilitate (15%), Testare & disciplină (10%).

**Referințe (APA, cu DOI)**

Hall, R. H., & Hanna, P. (2004). The impact of web page text‑background colour combinations on readability, retention, aesthetics, and behavioural intention. Behaviour & Information Technology, 23(3), 183–195. https://doi.org/10.1080/01449290410001669932

Tuch, A. N., Bargas‑Avila, J. A., Opwis, K., & Wilhelm, F. H. (2009). Visual complexity of websites: Effects on users’ experience, physiology, performance, and memory. International Journal of Human‑Computer Studies, 67(9), 703–715. https://doi.org/10.1016/j.ijhcs.2009.04.002

Seckler, M., Heinz, S., Bargas‑Avila, J. A., Opwis, K., & Tuch, A. N. (2014). Designing usable web forms: Empirical evaluation of web form improvement guidelines. Proceedings of CHI 2014. https://doi.org/10.1145/2556288.2557265

Hick, W. E. (1952). On the rate of gain of information. Quarterly Journal of Experimental Psychology, 4(1), 11–26. https://doi.org/10.1080/17470215208416600

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, 47(6), 381–391. https://doi.org/10.1037/h0055392

Sharma, G., Wu, W., & Dalal, E. N. (2005). The CIEDE2000 color‑difference formula. Color Research & Application, 30(1), 21–30. https://doi.org/10.1002/col.20070