**Seminarul 10 — React: componente și JSX — Partea 3 (Proiecte/teme, extins)**

\*\*Structură generală:\*\* 45 proiecte (15 × L1, 15 × L2, 15 × L3). Fiecare are `starter/` + `solution/`, teste Vitest & Jest, README cu cerințe și pași.

# L1 (Fundamental)

## L1-P01 — JSX Basics: Render a Title

\*\*Tema:\*\* JSX Basics: Render a Title — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru JSX Basics: Render a Title: implementare minimă care satisface testele (render <h1 data-testid='title'>JSX Basics: Render a Title</h1> + comportamentul cerut).

## L1-P02 — Props Drilling Intro: ClubCard receives props

\*\*Tema:\*\* Props Drilling Intro: ClubCard receives props — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Props Drilling Intro: ClubCard receives props: implementare minimă care satisface testele (render <h1 data-testid='title'>Props Drilling Intro: ClubCard receives props</h1> + comportamentul cerut).

## L1-P03 — State Basics: Counter Button

\*\*Tema:\*\* State Basics: Counter Button — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru State Basics: Counter Button: implementare minimă care satisface testele (render <h1 data-testid='title'>State Basics: Counter Button</h1> + comportamentul cerut).

## L1-P04 — Controlled Input: SearchBox

\*\*Tema:\*\* Controlled Input: SearchBox — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Controlled Input: SearchBox: implementare minimă care satisface testele (render <h1 data-testid='title'>Controlled Input: SearchBox</h1> + comportamentul cerut).

## L1-P05 — Lists & Keys: ClubList

\*\*Tema:\*\* Lists & Keys: ClubList — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Lists & Keys: ClubList: implementare minimă care satisface testele (render <h1 data-testid='title'>Lists & Keys: ClubList</h1> + comportamentul cerut).

## L1-P06 — Conditional Rendering: Empty State

\*\*Tema:\*\* Conditional Rendering: Empty State — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Conditional Rendering: Empty State: implementare minimă care satisface testele (render <h1 data-testid='title'>Conditional Rendering: Empty State</h1> + comportamentul cerut).

## L1-P07 — Composition: Card & Stack

\*\*Tema:\*\* Composition: Card & Stack — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Composition: Card & Stack: implementare minimă care satisface testele (render <h1 data-testid='title'>Composition: Card & Stack</h1> + comportamentul cerut).

## L1-P08 — Events: Toggle Details

\*\*Tema:\*\* Events: Toggle Details — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Events: Toggle Details: implementare minimă care satisface testele (render <h1 data-testid='title'>Events: Toggle Details</h1> + comportamentul cerut).

## L1-P09 — Derived UI: Computed Badge

\*\*Tema:\*\* Derived UI: Computed Badge — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Derived UI: Computed Badge: implementare minimă care satisface testele (render <h1 data-testid='title'>Derived UI: Computed Badge</h1> + comportamentul cerut).

## L1-P10 — Form Minimal: Add Club

\*\*Tema:\*\* Form Minimal: Add Club — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Form Minimal: Add Club: implementare minimă care satisface testele (render <h1 data-testid='title'>Form Minimal: Add Club</h1> + comportamentul cerut).

## L1-P11 — Immutability: Add/Remove Items

\*\*Tema:\*\* Immutability: Add/Remove Items — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Immutability: Add/Remove Items: implementare minimă care satisface testele (render <h1 data-testid='title'>Immutability: Add/Remove Items</h1> + comportamentul cerut).

## L1-P12 — Accessibility: Roles & Labels

\*\*Tema:\*\* Accessibility: Roles & Labels — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Accessibility: Roles & Labels: implementare minimă care satisface testele (render <h1 data-testid='title'>Accessibility: Roles & Labels</h1> + comportamentul cerut).

## L1-P13 — Refactor: Custom Hook Extraction

\*\*Tema:\*\* Refactor: Custom Hook Extraction — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Refactor: Custom Hook Extraction: implementare minimă care satisface testele (render <h1 data-testid='title'>Refactor: Custom Hook Extraction</h1> + comportamentul cerut).

## L1-P14 — Testing RTL: Queries by Role

\*\*Tema:\*\* Testing RTL: Queries by Role — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Testing RTL: Queries by Role: implementare minimă care satisface testele (render <h1 data-testid='title'>Testing RTL: Queries by Role</h1> + comportamentul cerut).

## L1-P15 — Snapshot-lite: toBeInTheDocument checks

\*\*Tema:\*\* Snapshot-lite: toBeInTheDocument checks — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Snapshot-lite: toBeInTheDocument checks: implementare minimă care satisface testele (render <h1 data-testid='title'>Snapshot-lite: toBeInTheDocument checks</h1> + comportamentul cerut).

# L2 (Intermediar)

## L2-P01 — Context Intro: Theme Switch

\*\*Tema:\*\* Context Intro: Theme Switch — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Context Intro: Theme Switch: implementare minimă care satisface testele (render <h1 data-testid='title'>Context Intro: Theme Switch</h1> + comportamentul cerut).

## L2-P02 — Custom Hooks: useFilteredClubs

\*\*Tema:\*\* Custom Hooks: useFilteredClubs — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Custom Hooks: useFilteredClubs: implementare minimă care satisface testele (render <h1 data-testid='title'>Custom Hooks: useFilteredClubs</h1> + comportamentul cerut).

## L2-P03 — Compound Components: Card.Header/Card.Body

\*\*Tema:\*\* Compound Components: Card.Header/Card.Body — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Compound Components: Card.Header/Card.Body: implementare minimă care satisface testele (render <h1 data-testid='title'>Compound Components: Card.Header/Card.Body</h1> + comportamentul cerut).

## L2-P04 — Keys Stability: Reorder List

\*\*Tema:\*\* Keys Stability: Reorder List — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Keys Stability: Reorder List: implementare minimă care satisface testele (render <h1 data-testid='title'>Keys Stability: Reorder List</h1> + comportamentul cerut).

## L2-P05 — Conditional Branching: Guarded Returns

\*\*Tema:\*\* Conditional Branching: Guarded Returns — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Conditional Branching: Guarded Returns: implementare minimă care satisface testele (render <h1 data-testid='title'>Conditional Branching: Guarded Returns</h1> + comportamentul cerut).

## L2-P06 — Form Validation: Minimal Rules

\*\*Tema:\*\* Form Validation: Minimal Rules — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Form Validation: Minimal Rules: implementare minimă care satisface testele (render <h1 data-testid='title'>Form Validation: Minimal Rules</h1> + comportamentul cerut).

## L2-P07 — Performance Primer: React.memo

\*\*Tema:\*\* Performance Primer: React.memo — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Performance Primer: React.memo: implementare minimă care satisface testele (render <h1 data-testid='title'>Performance Primer: React.memo</h1> + comportamentul cerut).

## L2-P08 — Deferred Updates: startTransition

\*\*Tema:\*\* Deferred Updates: startTransition — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Deferred Updates: startTransition: implementare minimă care satisface testele (render <h1 data-testid='title'>Deferred Updates: startTransition</h1> + comportamentul cerut).

## L2-P09 — Code Splitting: React.lazy

\*\*Tema:\*\* Code Splitting: React.lazy — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Code Splitting: React.lazy: implementare minimă care satisface testele (render <h1 data-testid='title'>Code Splitting: React.lazy</h1> + comportamentul cerut).

## L2-P10 — Portals Intro: Modal

\*\*Tema:\*\* Portals Intro: Modal — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Portals Intro: Modal: implementare minimă care satisface testele (render <h1 data-testid='title'>Portals Intro: Modal</h1> + comportamentul cerut).

## L2-P11 — Controlled vs Uncontrolled: Trade-offs

\*\*Tema:\*\* Controlled vs Uncontrolled: Trade-offs — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Controlled vs Uncontrolled: Trade-offs: implementare minimă care satisface testele (render <h1 data-testid='title'>Controlled vs Uncontrolled: Trade-offs</h1> + comportamentul cerut).

## L2-P12 — Error Boundary (concept & stub)

\*\*Tema:\*\* Error Boundary (concept & stub) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Error Boundary (concept & stub): implementare minimă care satisface testele (render <h1 data-testid='title'>Error Boundary (concept & stub)</h1> + comportamentul cerut).

## L2-P13 — A11y Pass: Keyboard Navigation

\*\*Tema:\*\* A11y Pass: Keyboard Navigation — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru A11y Pass: Keyboard Navigation: implementare minimă care satisface testele (render <h1 data-testid='title'>A11y Pass: Keyboard Navigation</h1> + comportamentul cerut).

## L2-P14 — List Item Identity: Preserving State

\*\*Tema:\*\* List Item Identity: Preserving State — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru List Item Identity: Preserving State: implementare minimă care satisface testele (render <h1 data-testid='title'>List Item Identity: Preserving State</h1> + comportamentul cerut).

## L2-P15 — Testing Library Advanced Queries

\*\*Tema:\*\* Testing Library Advanced Queries — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Testing Library Advanced Queries: implementare minimă care satisface testele (render <h1 data-testid='title'>Testing Library Advanced Queries</h1> + comportamentul cerut).

# L3 (Avansat)

## L3-P01 — State Orchestration: useReducer

\*\*Tema:\*\* State Orchestration: useReducer — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru State Orchestration: useReducer: implementare minimă care satisface testele (render <h1 data-testid='title'>State Orchestration: useReducer</h1> + comportamentul cerut).

## L3-P02 — Optimistic UI (client-only)

\*\*Tema:\*\* Optimistic UI (client-only) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Optimistic UI (client-only): implementare minimă care satisface testele (render <h1 data-testid='title'>Optimistic UI (client-only)</h1> + comportamentul cerut).

## L3-P03 — Virtualized List (conceptual stub)

\*\*Tema:\*\* Virtualized List (conceptual stub) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Virtualized List (conceptual stub): implementare minimă care satisface testele (render <h1 data-testid='title'>Virtualized List (conceptual stub)</h1> + comportamentul cerut).

## L3-P04 — Async Transition for Filtering

\*\*Tema:\*\* Async Transition for Filtering — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Async Transition for Filtering: implementare minimă care satisface testele (render <h1 data-testid='title'>Async Transition for Filtering</h1> + comportamentul cerut).

## L3-P05 — Wizard Form (multi-step state)

\*\*Tema:\*\* Wizard Form (multi-step state) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Wizard Form (multi-step state): implementare minimă care satisface testele (render <h1 data-testid='title'>Wizard Form (multi-step state)</h1> + comportamentul cerut).

## L3-P06 — Compound Pattern Advanced

\*\*Tema:\*\* Compound Pattern Advanced — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Compound Pattern Advanced: implementare minimă care satisface testele (render <h1 data-testid='title'>Compound Pattern Advanced</h1> + comportamentul cerut).

## L3-P07 — Design System Primitives

\*\*Tema:\*\* Design System Primitives — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Design System Primitives: implementare minimă care satisface testele (render <h1 data-testid='title'>Design System Primitives</h1> + comportamentul cerut).

## L3-P08 — Skeleton Loading UI

\*\*Tema:\*\* Skeleton Loading UI — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Skeleton Loading UI: implementare minimă care satisface testele (render <h1 data-testid='title'>Skeleton Loading UI</h1> + comportamentul cerut).

## L3-P09 — Reducer + Context Integration

\*\*Tema:\*\* Reducer + Context Integration — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Reducer + Context Integration: implementare minimă care satisface testele (render <h1 data-testid='title'>Reducer + Context Integration</h1> + comportamentul cerut).

## L3-P10 — Performance Profiling (concept)

\*\*Tema:\*\* Performance Profiling (concept) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Performance Profiling (concept): implementare minimă care satisface testele (render <h1 data-testid='title'>Performance Profiling (concept)</h1> + comportamentul cerut).

## L3-P11 — Error Boundaries Advanced

\*\*Tema:\*\* Error Boundaries Advanced — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Error Boundaries Advanced: implementare minimă care satisface testele (render <h1 data-testid='title'>Error Boundaries Advanced</h1> + comportamentul cerut).

## L3-P12 — Children as Function (render props demo)

\*\*Tema:\*\* Children as Function (render props demo) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Children as Function (render props demo): implementare minimă care satisface testele (render <h1 data-testid='title'>Children as Function (render props demo)</h1> + comportamentul cerut).

## L3-P13 — Testability by Design (fixtures)

\*\*Tema:\*\* Testability by Design (fixtures) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Testability by Design (fixtures): implementare minimă care satisface testele (render <h1 data-testid='title'>Testability by Design (fixtures)</h1> + comportamentul cerut).

## L3-P14 — Story-like Test Cases (RTL)

\*\*Tema:\*\* Story-like Test Cases (RTL) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Story-like Test Cases (RTL): implementare minimă care satisface testele (render <h1 data-testid='title'>Story-like Test Cases (RTL)</h1> + comportamentul cerut).

## L3-P15 — Upgrade Notes 18→19 (guard rails)

\*\*Tema:\*\* Upgrade Notes 18→19 (guard rails) — specificație: implementați comportamentul descris și treceți testele RTL (Vitest & Jest).

\*\*Cerințe minimale:\*\* 1) `<h1 data-testid="title">` cu titlul exact; 2) `input` controlat (`aria-label="search-input"`).

\*\*Teste:\*\* suite echivalente RTL (Vitest & Jest) verifică titlul și inputul; soluția oferă un `startTransition` la `onChange`.

\*\*Soluție (sumar):\*\* Soluție orientativă pentru Upgrade Notes 18→19 (guard rails): implementare minimă care satisface testele (render <h1 data-testid='title'>Upgrade Notes 18→19 (guard rails)</h1> + comportamentul cerut).

---  
\*\*Fragmente comune (o singură dată în document):\*\*

// tests/vitest/App.test.tsx  
import { render, screen } from '@testing-library/react'  
import userEvent from '@testing-library/user-event'  
import App from '../../src/App'  
  
test('renders project title', () => {  
 render(<App />)  
 expect(screen.getByTestId('title')).toBeInTheDocument()  
})  
  
test('has a controlled search input (solution will pass)', async () => {  
 render(<App />)  
 const user = userEvent.setup()  
 const input = screen.getByLabelText('search-input')  
 await user.type(input, 'abc')  
 expect(input).toBeInTheDocument()  
})

// src/App.tsx (solution skeleton)  
import React, { useState, startTransition } from 'react'  
  
export default function App(){   
 const [query, setQuery] = useState('')  
 return (  
 <main className="container">  
 <h1 data-testid="title">PROJECT\_TITLE\_SAMPLE</h1>  
 <label aria-label="search">  
 <span>Search:</span>  
 <input aria-label="search-input" value={query} onChange={e => startTransition(() => setQuery(e.target.value))} />  
 </label>  
 <p className="card">Solution skeleton — query length: {query.length}</p>  
 </main>  
 )  
}