

Crafting

Design Systems

that looks good, and works good!



@kothariji

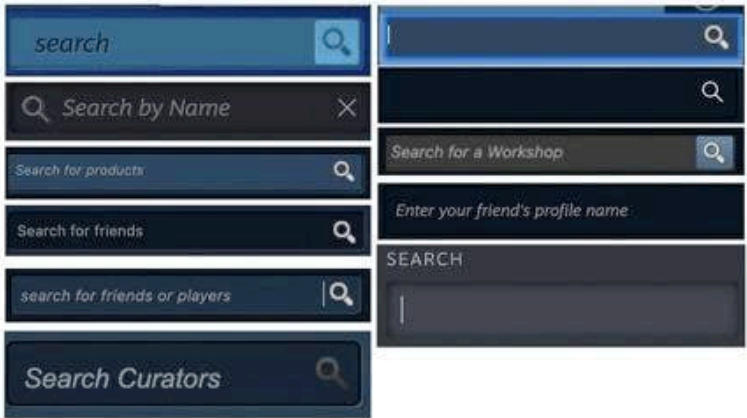
Dhruv Kothari

Ex- Zomato, Upraised

Question

**Why do we need
Design Systems
in first place?**

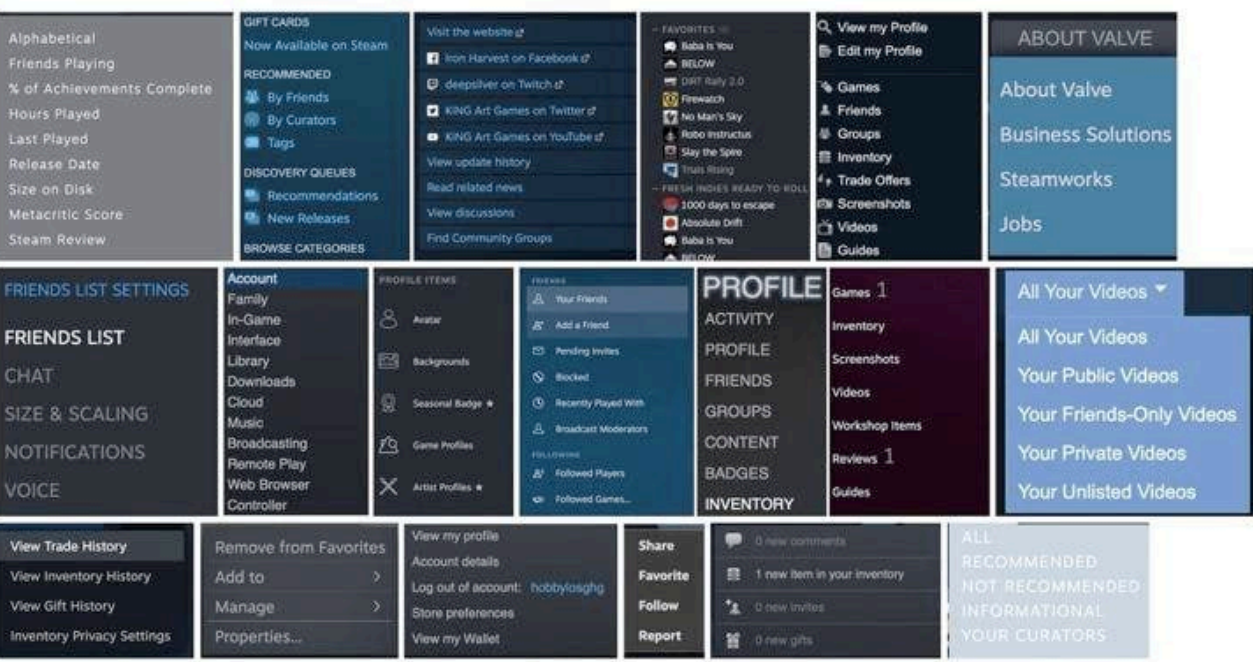
Search bars of Steam:



Tab menus of steam:



Drop-down/List menus of Steam:



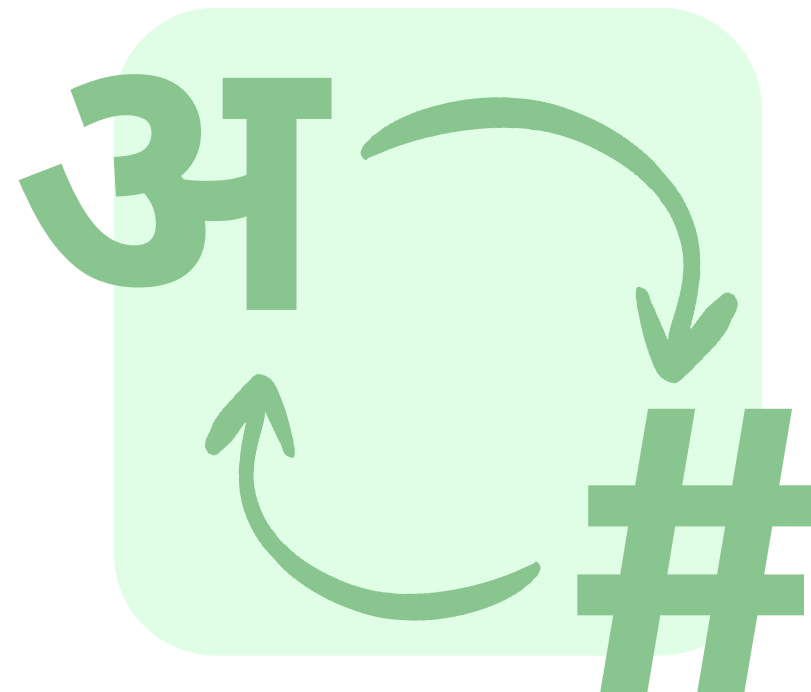
Buttons of steam:



Grammar prevents chaos. So does a good design system.

Design Systems as Grammar

Alphabets & Words



Tokens & Components

Colors, spacing, typography — these are your alphabets.

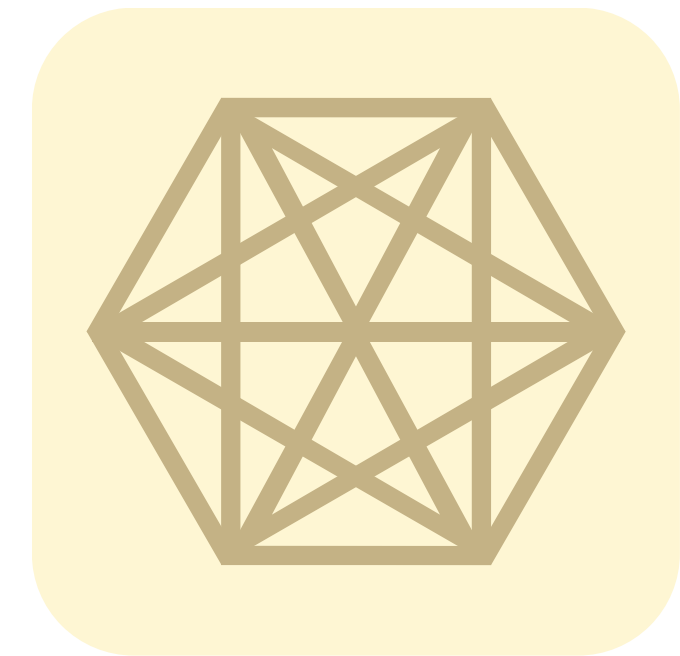
Sentences



Composition

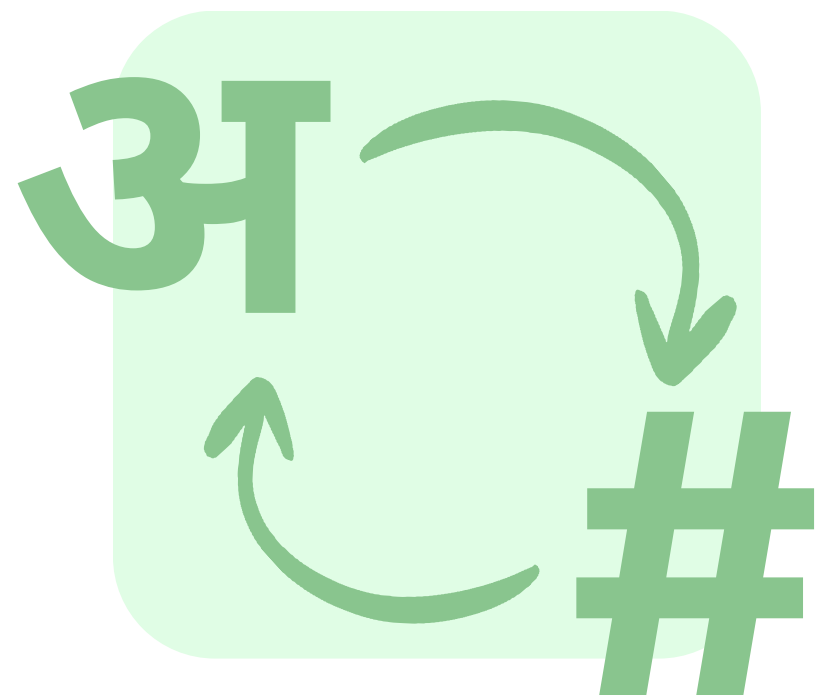
Buttons, inputs, forms — They express intent, not just style.

Grammar



Guidelines & Laws

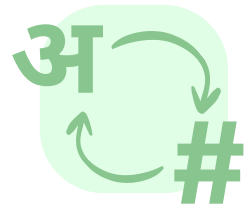
Principles of UX, Standardization, Internationalization, and Accessibility



Alphabets and Words

Tokens and Components

Tokens evolve from raw values → core tokens → semantic tokens → component tokens.
They define multiple attributes — color, typography, spacing, sizing, radius, opacity, stroke width, and effects.



Alphabets and Words

Tokens and Components

Gray

colors.coolGray



50	100	200	300	400	500	600	700	800	900
#F9FAFB	#F3F4F6	#E5E7EB	#D1D5DB	#9CA3AF	#6B7280	#4B5563	#374151	#1F2937	#111827

Red

colors.red



50	100	200	300	400	500	600	700	800	900
#FEF2F2	#FEE2E2	#FECACA	#FCA5A5	#F87171	#EF4444	#DC2626	#B91C1C	#991B1B	#7F1D1D

Yellow

colors.amber



50	100	200	300	400	500	600	700	800	900
#FFFBE5	#FEF3C7	#FDE68A	#FCD34D	#FBBF24	#F59E0B	#D97706	#B45309	#92400E	#78350F

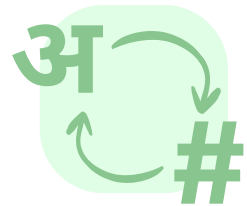
Green

colors.emerald



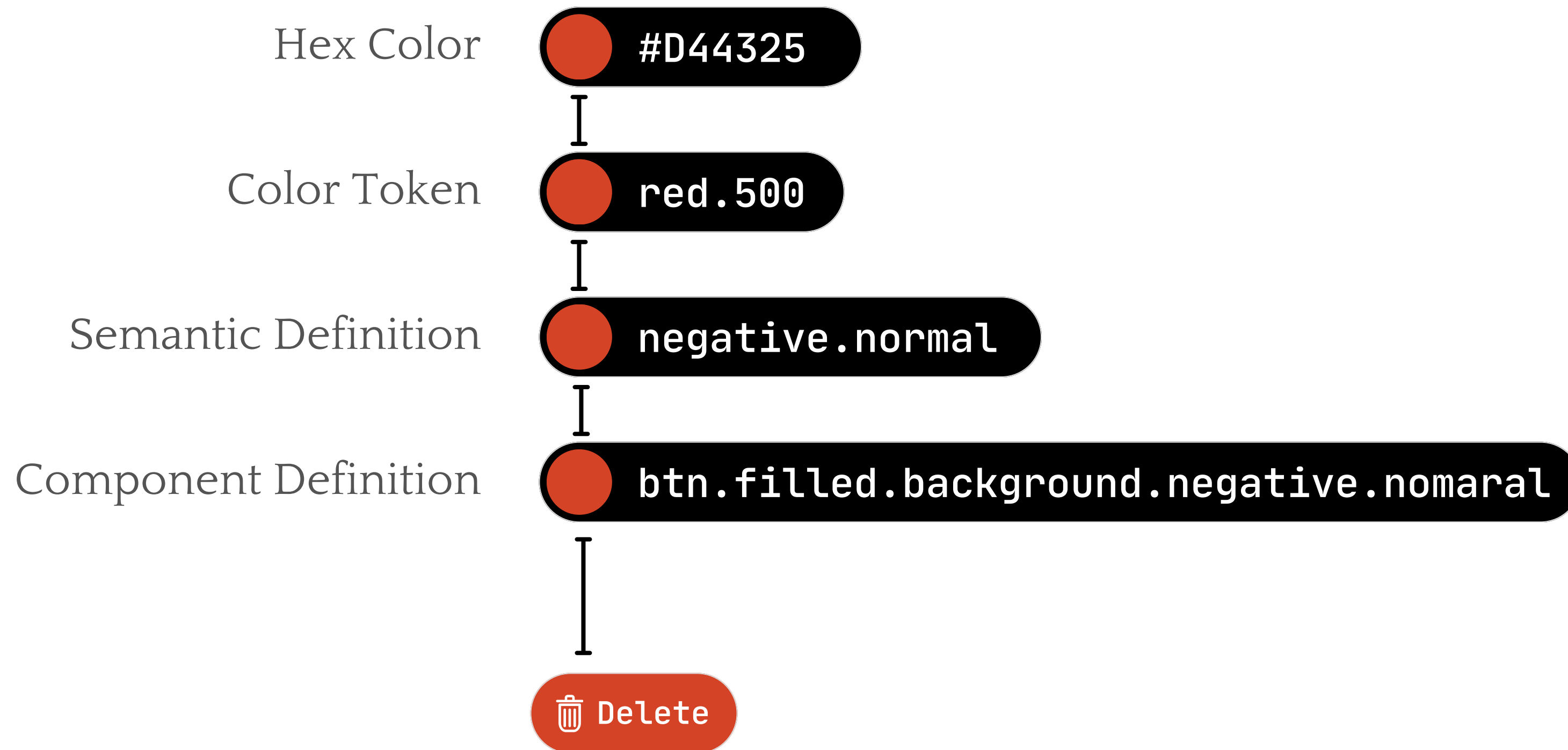
50	100	200	300	400	500	600	700	800	900
#ECFDF5	#D1FAE5	#A7F3D0	#6EE7B7	#34D399	#10B981	#059669	#047857	#065F46	#064E3B

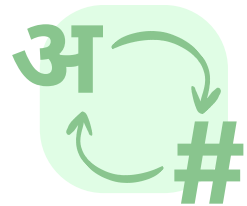
source - <https://ihatetomatoes-nextjs-101.vercel.app/post/customising-colors-tailwindcss>



Alphabets and Words

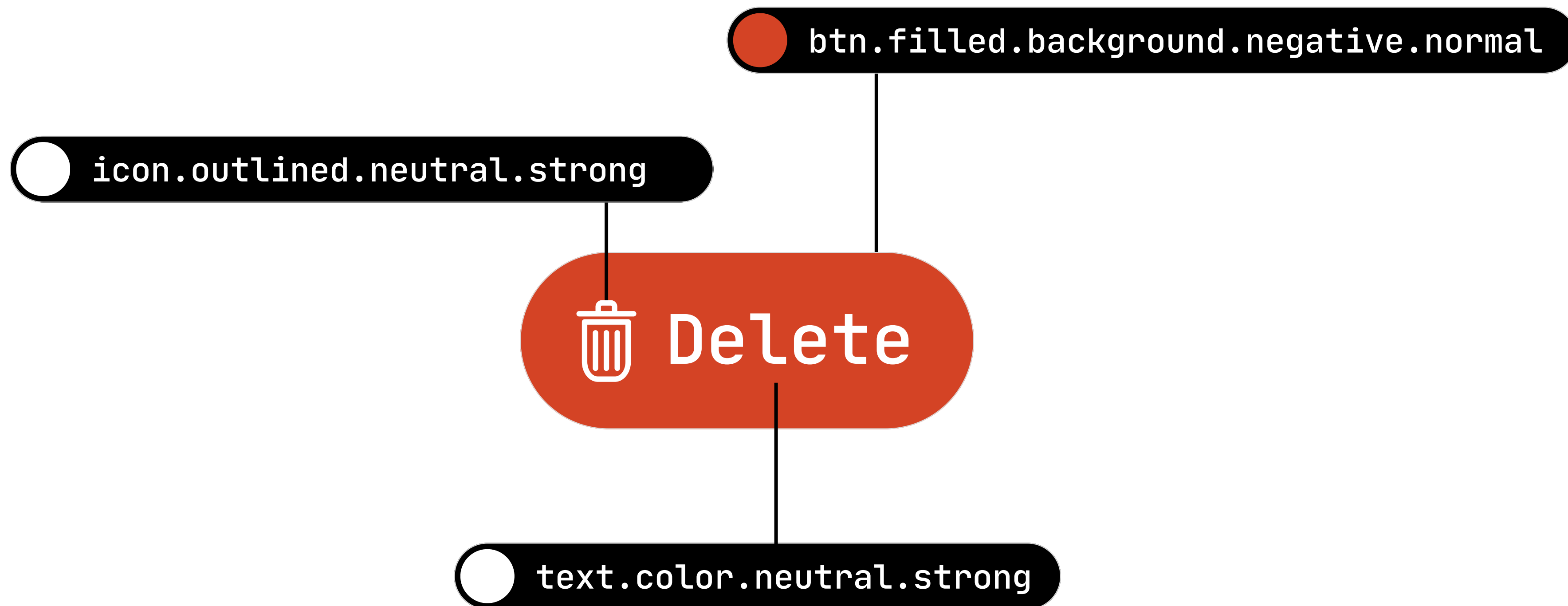
Tokens and Components

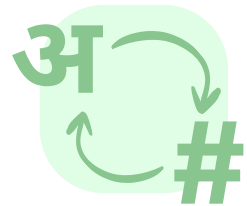




Alphabets and Words

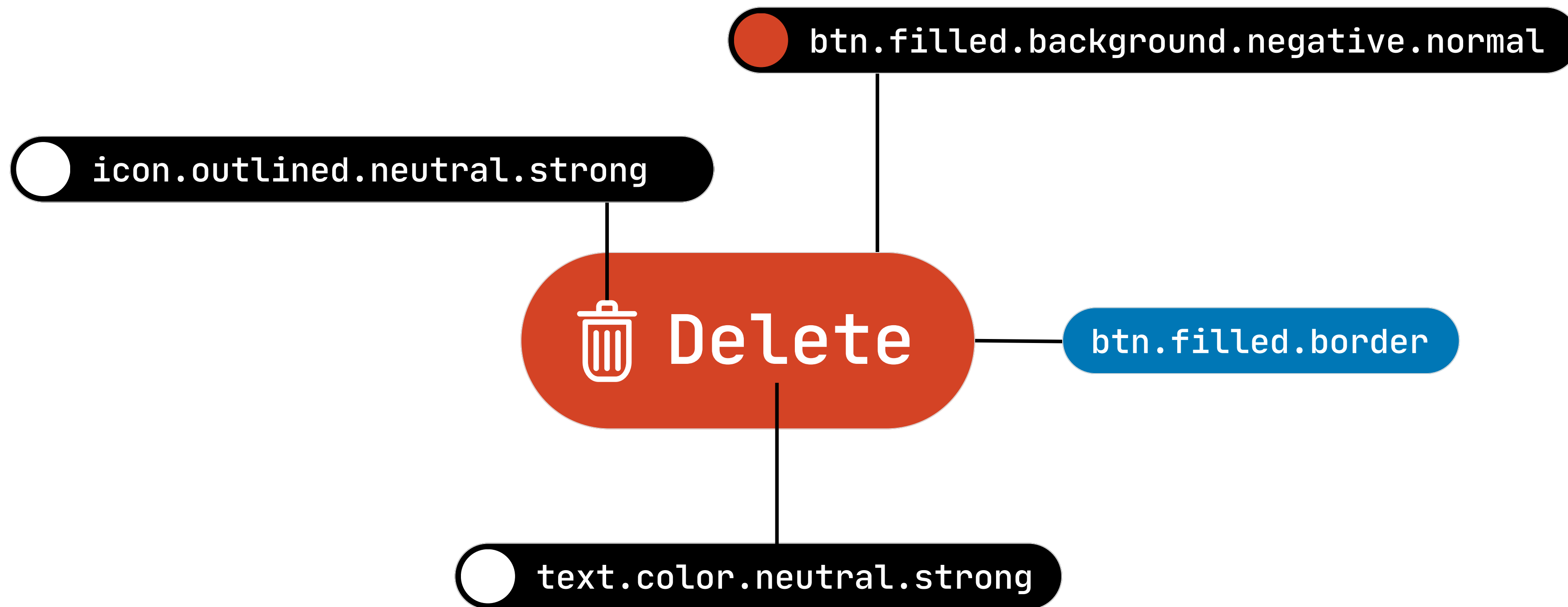
Tokens and Components

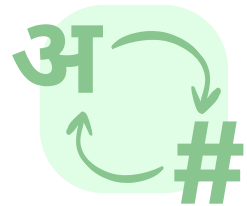




Alphabets and Words

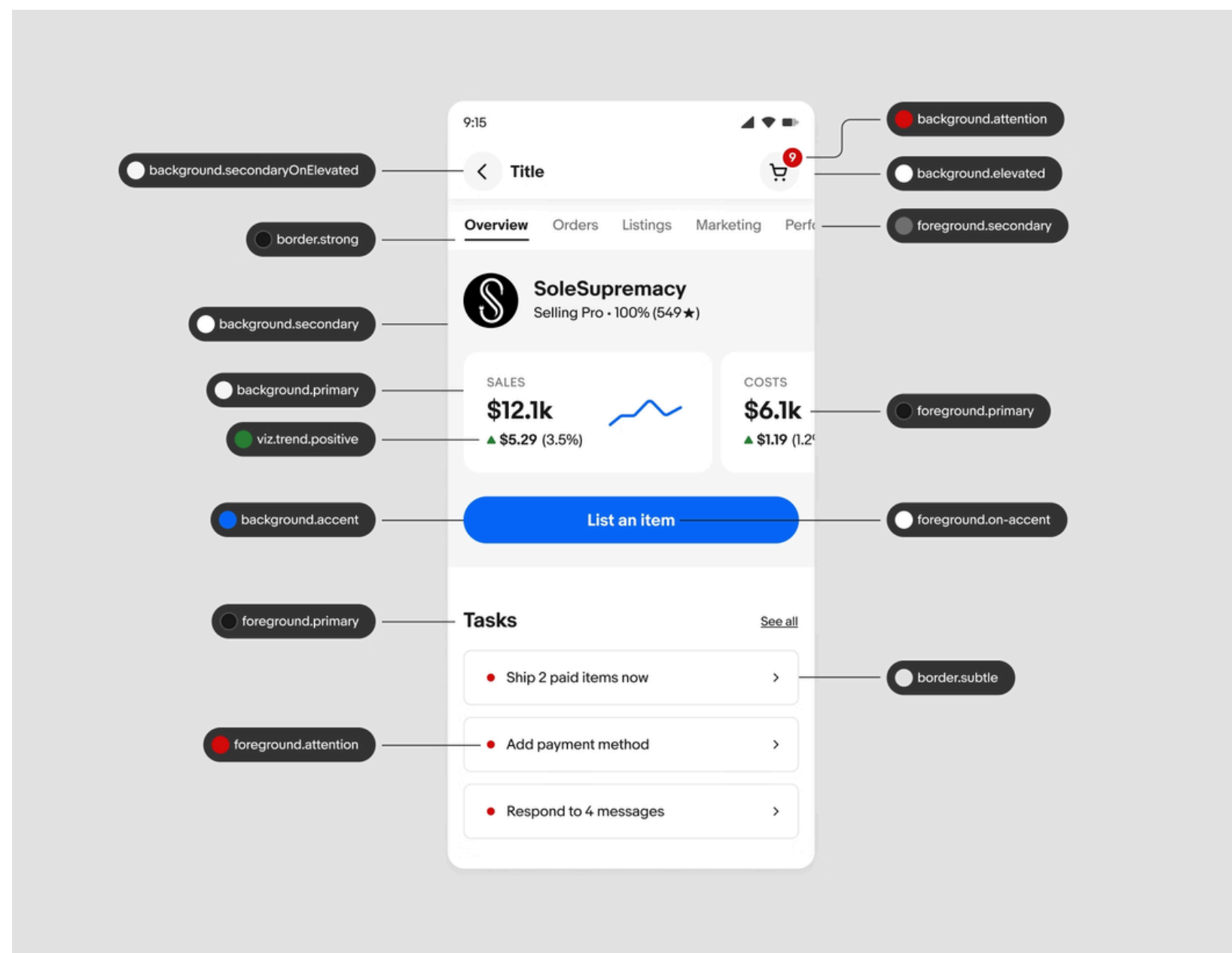
Tokens and Components





Alphabets and Words

Tokens and Components

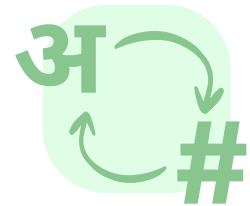


Source: eBay Playbook Design System (for educational reference)

Tokens

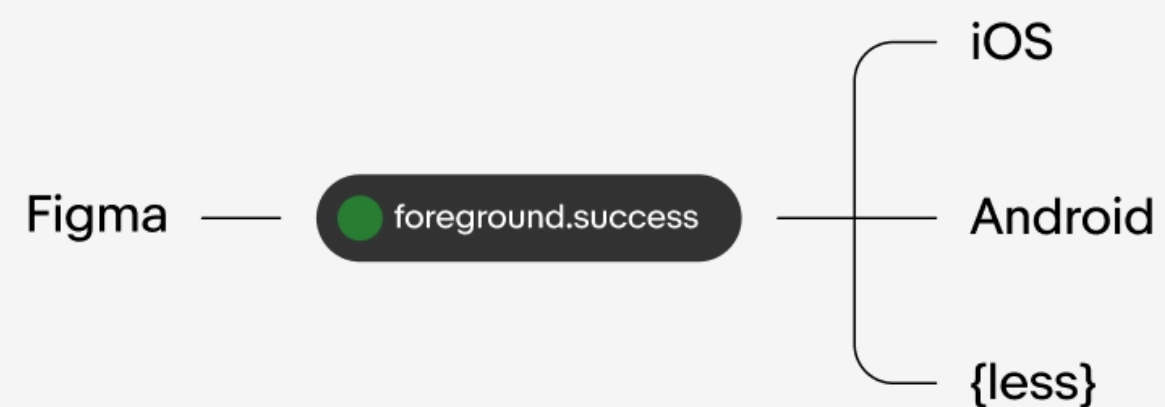
Page composition

Here's an example of how tokens are applied across a page. Background elements—such as containers and frames—layer beneath other components. Foreground elements, like text and icons, appear above these backgrounds. Finally, border elements encompass strokes and dividers that define and separate sections.



Alphabets and Words

Tokens and Components



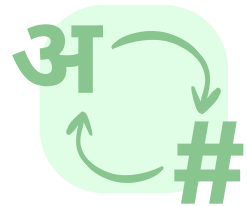
Source: eBay Playbook Design System (for educational reference)

Tokens

Platform agnostic

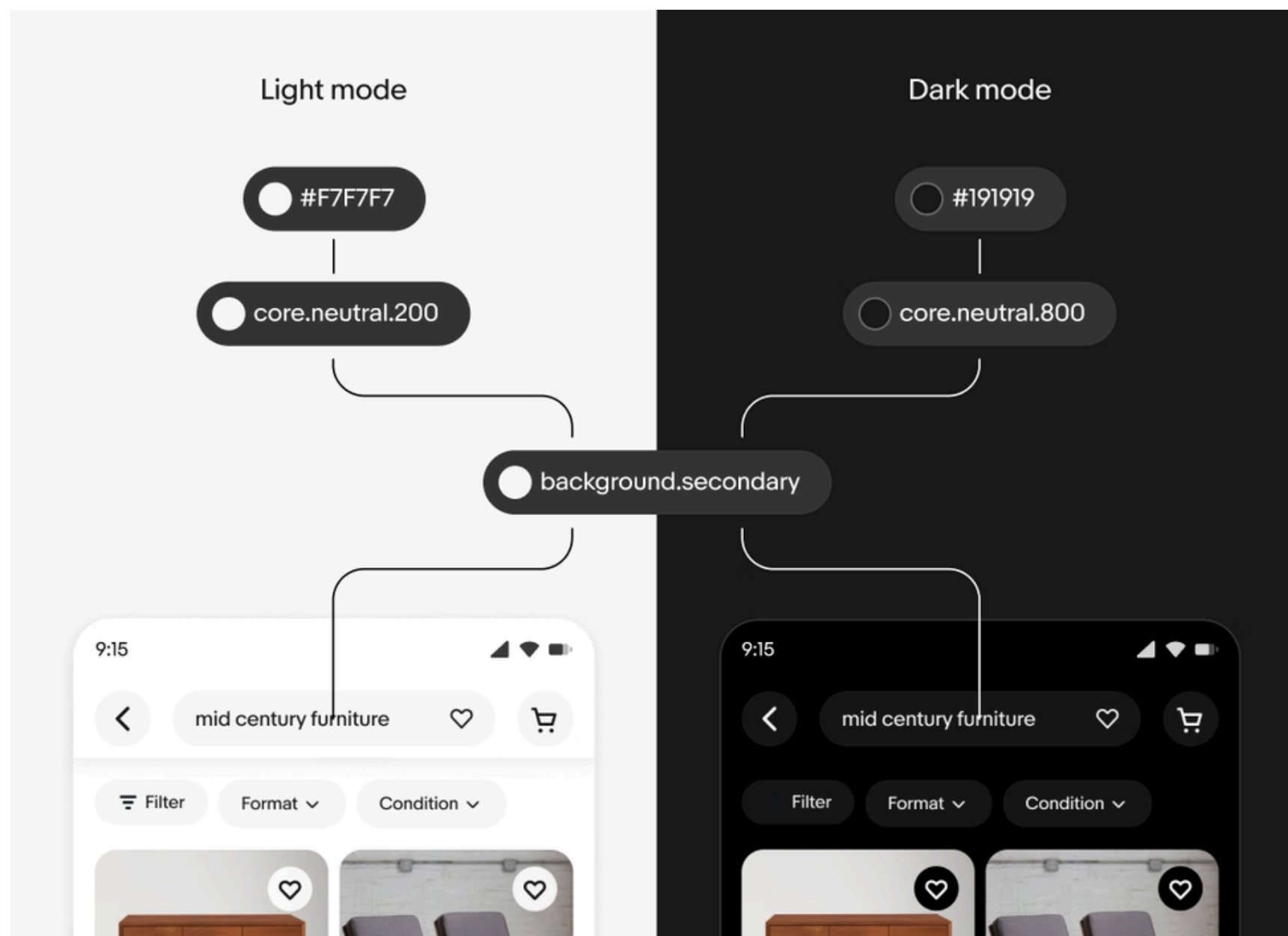
Tokens are platform-agnostic, with names remaining similar despite minor naming structure changes. The design library uses dots as separators and lowercase letters, but can be adapted to platform-specific conventions like dashes or camelCase.

Tokens streamline design documentation and make cross-team sharing easier, providing a common language for styles and intentions.



Alphabets and Words

Tokens and Components



Source: eBay Playbook Design System (for educational reference)

Tokens

Theming

A single token can point to different values depending on the context. The context can be a change in device theme, form factor, or accessibility settings.

When a context changes, the reference values are updated automatically and the changes cascade throughout the system.



Sentences

Composition

Composition refers to the practice of building complex user interfaces by combining multiple smaller, independent components. This approach is favored over inheritance for code reuse and creating modular, maintainable applications.



Sentences

Composition

Reusable components \neq composable systems.

A composable design system lets you build new experiences without creating new components.



Sentences

Composition

```
- <PrimaryButton />  
+ <Button variant="primary" />  
  
- <UserCard />  
+ <Card>  
+   <Avatar />  
+   <Text />  
+ </Card>
```

Composition

Focus on composition, not inheritance.

Don't bake context into components — let the parent define behavior and layout. Composition keeps complexity outside, flexibility inside.



Sentences

Composition



SelectDropdown.jsx

```
<InputWrapper label="Select Label">
  <InputBase>
    <LeftIcon>
      <SearchIcon />
    </LeftIcon>

    <InputField placeholder="Search..." />

    <RightIcon> <ClearIcon /> </RightIcon>
    <RightIcon> <ChevronIcon /> </RightIcon>
  </InputBase>

  <Popover>
    <MenuList>
      <MenuItem>Gurgaon</MenuItem>
      <MenuItem>Chennai</MenuItem>
      <MenuItem>Bangalore</MenuItem>
      <MenuItem selected>Pune</MenuItem>
    </MenuList>
  </Popover>
</InputWrapper>
```

Select Label

Q |

✕ ^

Gurgaon

Chennai

Bangalore

Pune ✓



Grammar — Rules

Guidelines & Laws

Design systems aren't just about consistency—they're about cognitive predictability. UX laws define how users think, react, and decide. When baked into a system's DNA, they save developers from reinventing logic every time. These principles—like Hick's, Fitts', and Jakob's laws—aren't just design wisdom; they're engineering shortcuts to faster, more usable, and accessible products.



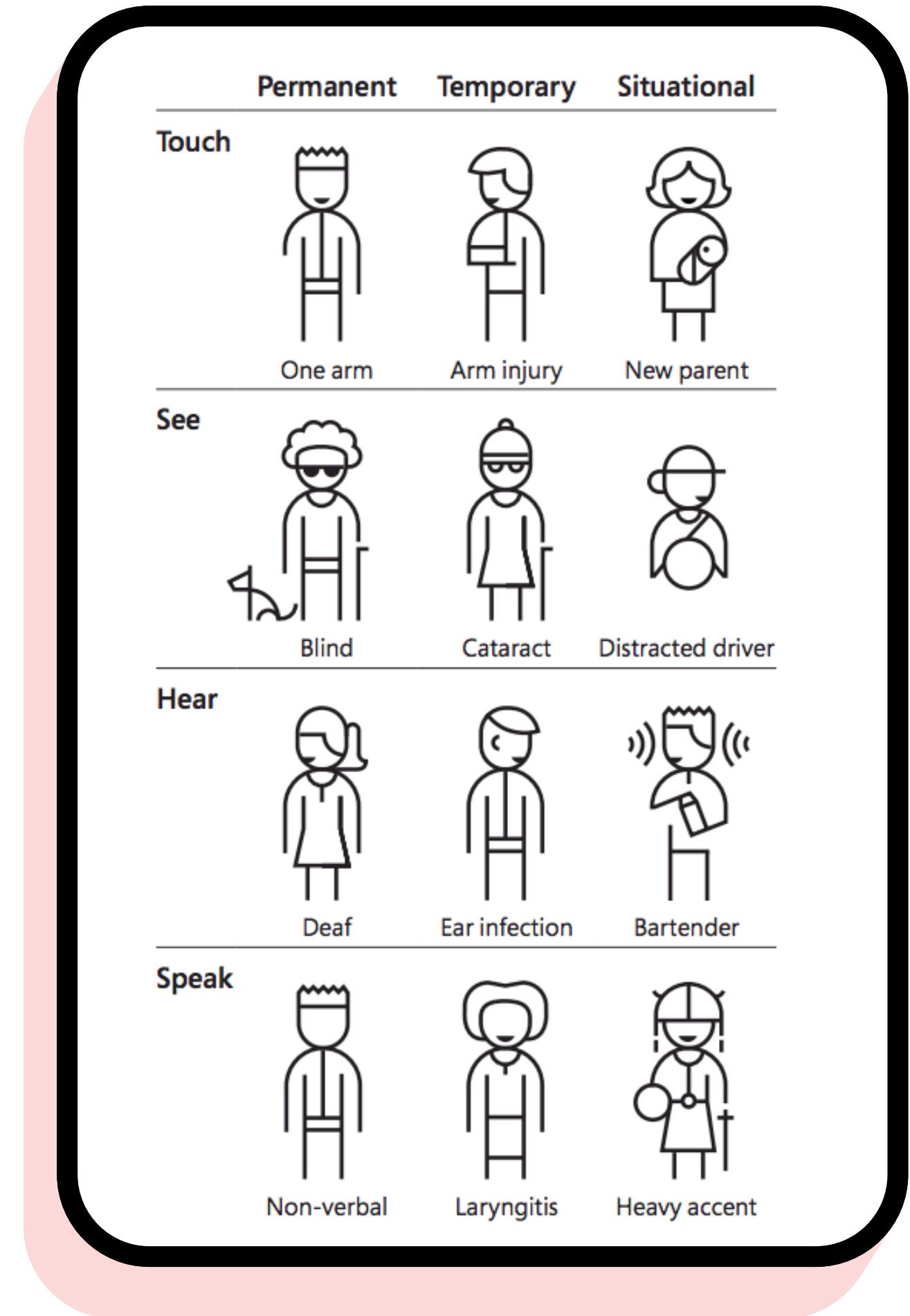
Grammar — Rules

Guidelines & Laws

Guidelines & Laws

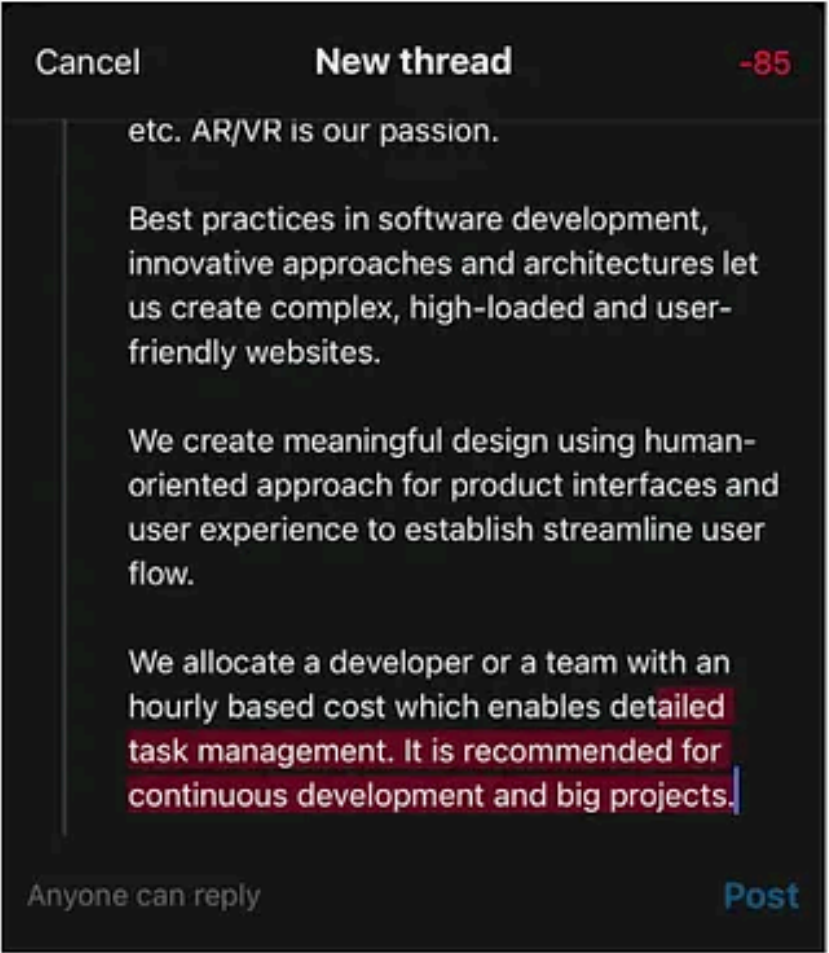
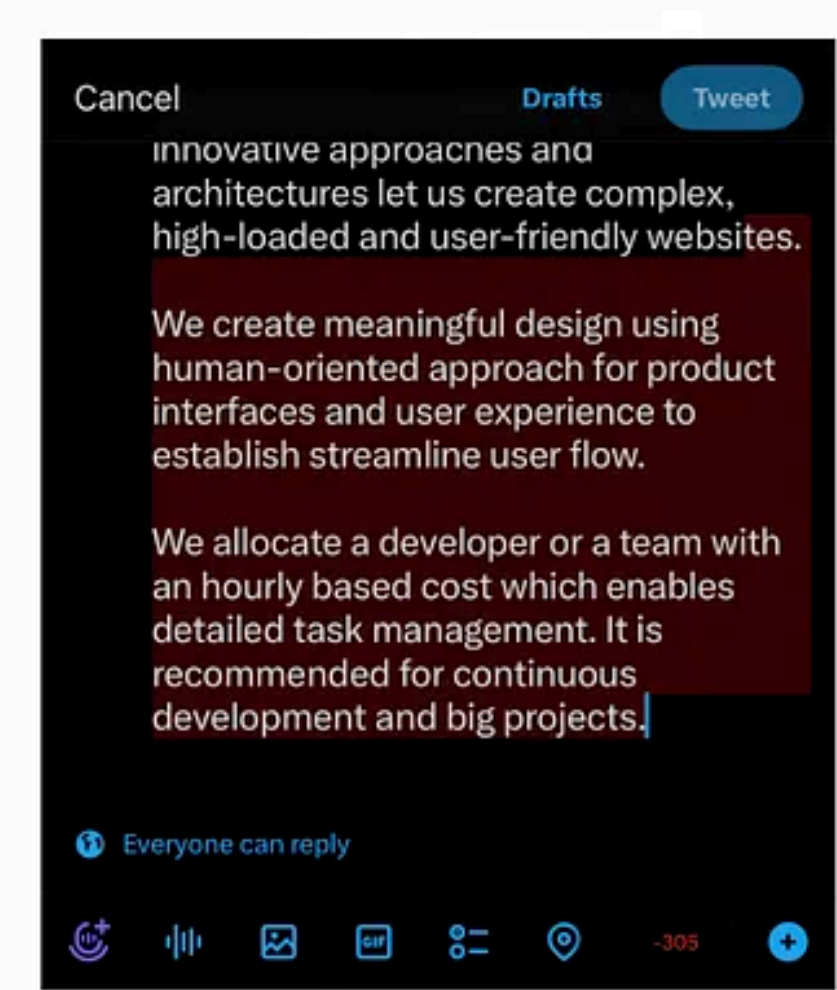
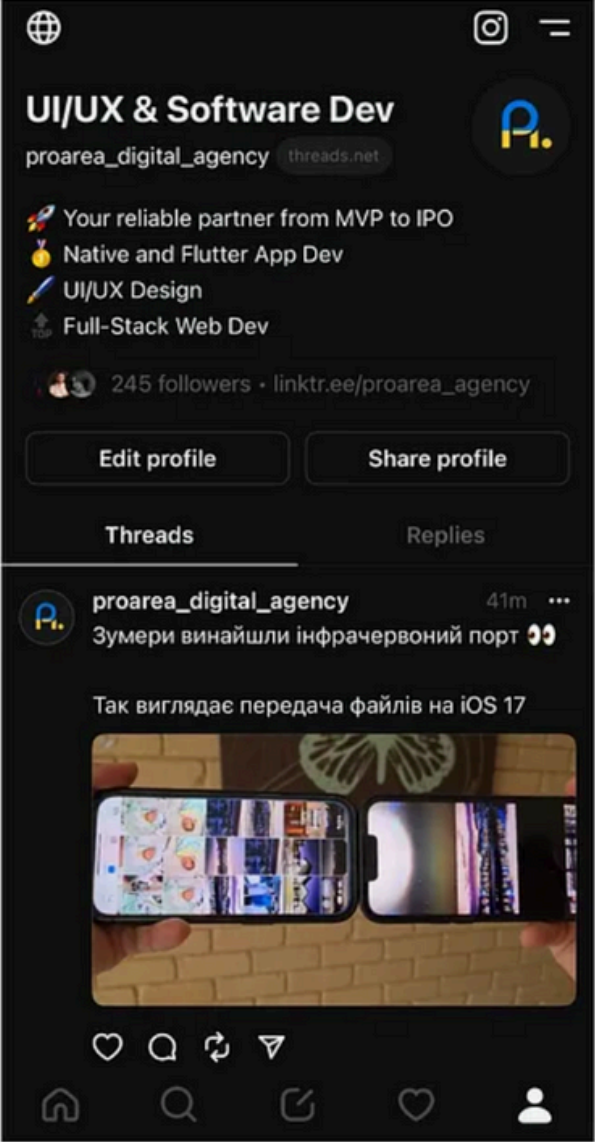
Solve for all!

Think about the end user, how are they gonna use your product?
What challenges can they face while using the product?



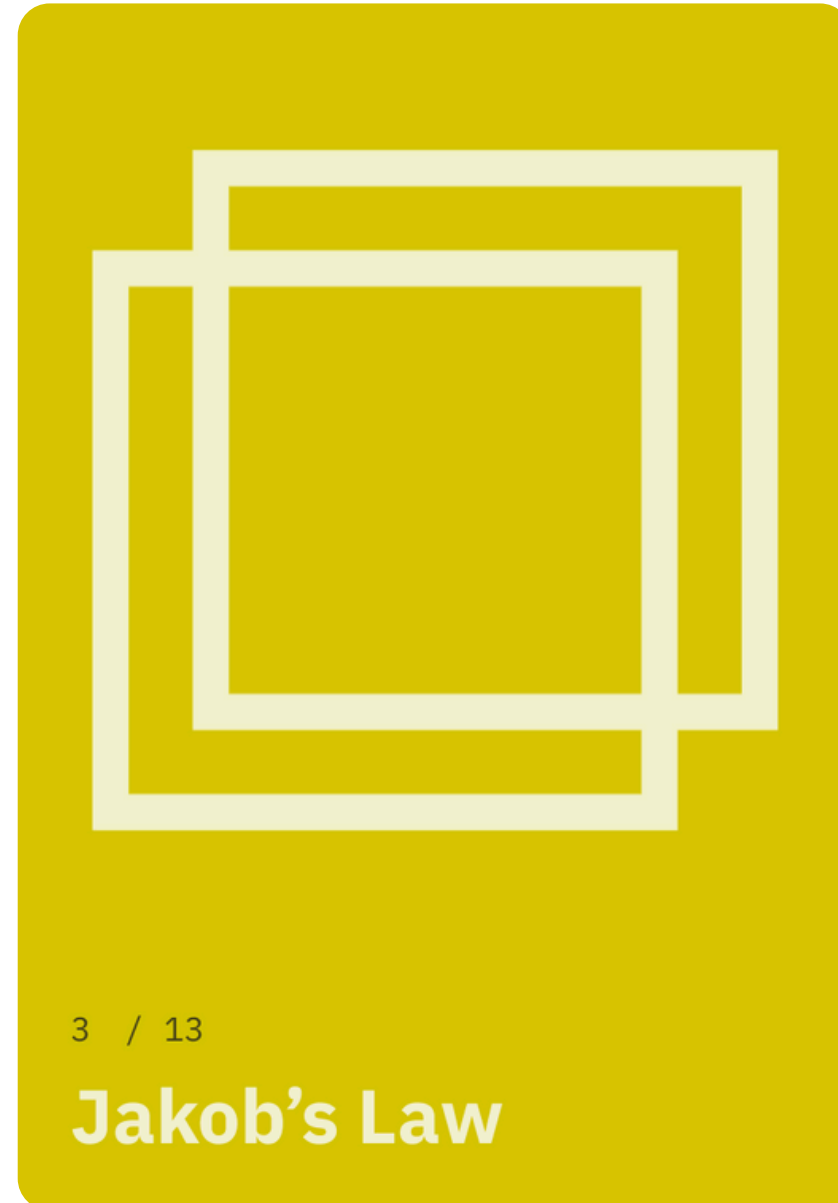


Guidelines & Laws





Guidelines & Laws



Laws of UX

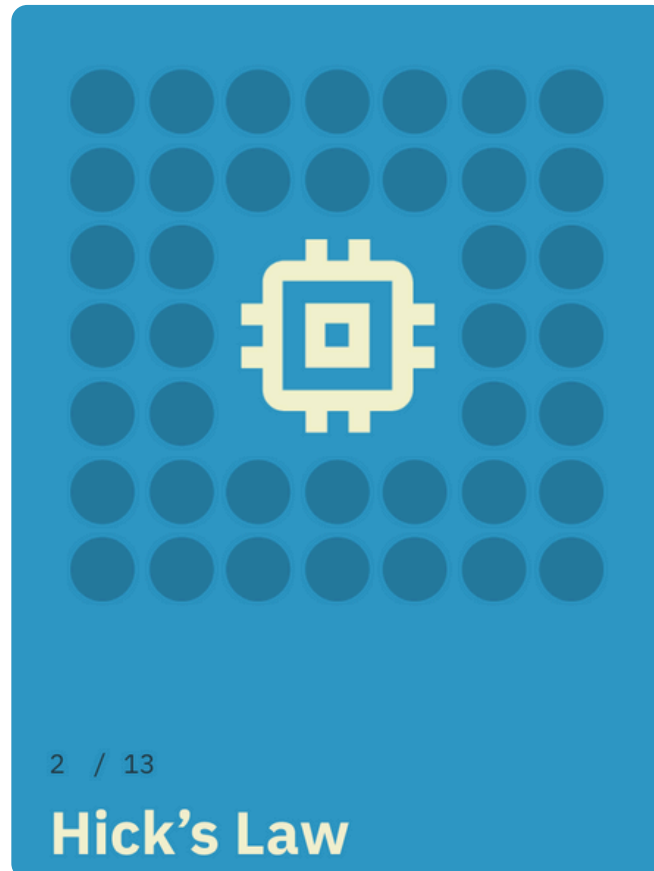
Jakob's Law — Familiar Feels Fast

Users spend most of their time on other apps, not yours. So, when Threads launched, its UI felt instantly familiar — because it mirrored Twitter's mental model. That wasn't laziness; it was Jakob's Law in action. Familiarity reduces cognitive load, accelerates adoption, and builds comfort faster than novelty ever could.

Threads didn't copy Twitter — it spoke the same visual language because fluency beats originality on day one.



Guidelines & Laws



Hick's Law

(more choices → slower decisions.)

Replace huge dropdowns with search/autocomplete or grouped choices.



Fitts' Law

(target size + distance for ease of tapping/clicking.)

Make interactive targets easy to hit — padding matters more than pixels.



Grammar — Rules

Guidelines & Laws

Guidelines & Laws

Design-system guidelines beyond laws

Design-system guidelines are the engineering contract that turns UX laws into reliable, repeatable product behavior — think tokens, motion presets, accessibility rules, and data standards (dates, numbers, currencies). Follow them and teams spend time building features, not debating pixels.



Grammar — Rules

Guidelines & Laws



Scan this QR code



Placeholders — Small Rules, Big UX Wins

1. Placeholders are never translated, they remain in the originally authored language

Name



नाम



Placeholders — Small Rules, Big UX Wins

2. Placeholders are temporary, we can't fully rely on them

Date of Birth

DD-MM-YYYY

Initial State

Date of Birth

25-

Typing State

Date of Birth (DD-MM-YYYY)

DD-MM-YYYY



Placeholders — Small Rules, Big UX Wins

Placeholder text's length is also limited to the width of the input it is contained in. In our responsive, mobile-first world, there stands a very good chance that important information could be truncated:

Password

Enter minimum...



Placeholders — Small Rules, Big UX Wins

Placeholders may look like pre-filled information and be skipped over. (Esp. in High Contrast Mode)
Can be too light-coloured to be legible; (bad for accessibility)

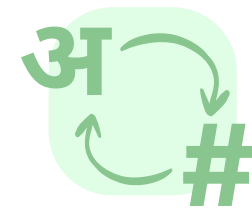


Guidelines & Laws

- ☐ Semantic HTML & ARIA where required.
- ☐ Keyboard nav + focus visible.
- ☐ Contrast checks passed. AA / AAA Accessibility Compliance Verification
- ☐ Dates/times in ISO on API; locale display at client.
- ☐ Spacing & scale: define a step system (4,8,16...) and enforce it.
- ☐ Motion preferences: respect prefers-reduced-motion.
- ☐ RTL & locale: test flow, mirroring, and numbers/units.
- ☐ Versioning & docs: changelogs, migration guides, and examples.
- ☐ Token usage only (no hardcoded colors/spacing).
- ☐ Storybook story + accessibility snapshot.
- ☐ Visual regression + unit tests.
- ☐ Deprecation note if you change token names.

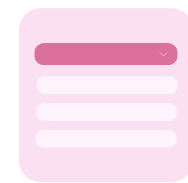
Simply put

**Teach your team the grammar —
then let them build usable products.**



Alphabets and Words

Tokens and Components



Sentences

Composition



Grammar — Rules

Guidelines & Laws

Closing

Ask me anything!

(except token naming, please).

Deck Link - <https://github.com/kothariji/react-india-pune-conf-oct-2025>

Design Systems | @kothariji