

AI UI Design Principles

A practical quality standard for developers using AI coding agents to design professional, consistent, usable interfaces.

The goal: stop accepting the first UI an AI generates. Give your agent a clear design standard, and make it check its own work.

For developers, designers and AI-assisted builders. Use this guide as an AGENT.md or project-level UI standard.

Why AI-generated UIs often look wrong

AI is very good at producing something that looks like a UI. That does not automatically mean it has produced a good interface.

Asked to build a page without strong design constraints, an agent reaches for familiar visual patterns: too many cards, unnecessary icons, gradients, badges, shadows, floating elements, repeated descriptions, inconsistent spacing, and sections added simply because there is empty space. The result is visually busy, inconsistent, and surprisingly difficult to use.

The problem is usually not that the AI cannot write the code. The problem is that it has not been given a sufficiently clear design decision framework.

This guide gives your agent that framework. Add it to your project as a persistent instruction file, then require the agent to consult it *both* before generating UI and after generating or refining it.

How to use this with your agent

1 Add the principles to a project-accessible instruction file

Create or update an AGENT.md, or another instruction file your agent reliably reads, and paste the principles into it.

2 Make them an explicit design requirement

They are not suggestions. They are the default UI quality standard for the project unless a specific product requirement overrides them.

3 Require a pre-build reference

Before generating a page, component, dashboard, form or flow, the agent should make its decisions from the principles rather than copying familiar AI-generated patterns.

4 Require a post-build audit

After generating or refining the UI, the agent returns to the same file, checks the implementation against the principles, and fixes violations before calling the work complete.

The principles

1 Core principle

- Every UI element must have a clear functional or usability purpose.
- If an element does not improve understanding, navigation, interaction, feedback or task completion, do not add it.

2 Simplicity

- Prefer the simplest interface that solves the user's problem.
- Do not add UI merely to make the page look more impressive.
- Avoid unnecessary visual complexity.
- Do not fill whitespace with decorative content.
- Prefer clarity over novelty.

3 Icons

- Icons are optional, not mandatory.
- Use them only where they improve recognition, navigation, status or action discoverability.
- Do not place an icon beside every heading, and do not use decorative icons.
- Do not use multiple icons where one or none is sufficient.
- Use one consistent icon library and visual style.

4 Descriptions

- Do not add a description beneath every heading, a page title should normally stand on its own.
- Add descriptions only when they carry information the user actually needs.
- Avoid generic lines such as "Manage your settings and preferences."
- Do not repeat information the UI already communicates.
- Use concise helper text only where clarification is necessary.

5 Cards

- Do not place every section inside a card.
- Use cards only where they create meaningful grouping or separation.
- Prefer simple sections, dividers, spacing and typography when they are sufficient.
- Avoid nested cards unless there is a strong UX reason.

6 Badges

- Use badges only for meaningful status, category or state.
- Do not use badges as decoration.
- Do not turn ordinary text into a badge unnecessarily.

7 Decoration

- Avoid unnecessary gradients, illustrations, background shapes, floating elements, decorative icons, excessive shadows and borders, and animated effects.
- Use decoration only where it contributes to the product experience.

8 Layout

- Establish a clear visual hierarchy.
- Use whitespace intentionally.
- Group related information; separate unrelated information.
- Keep primary actions visually obvious.
- Avoid unnecessary sections.
- Maintain consistent spacing and alignment.

9 Information hierarchy

- Prioritise in this order: what the user needs to know, what the user needs to do, additional context, then secondary or advanced options.
- Use progressive disclosure for advanced or rarely used functionality.

10 Forms

- Keep forms focused and easy to scan; group related fields.
- Avoid unnecessary helper text.
- Use sensible defaults and inline validation.
- Clearly distinguish required and optional fields.
- Do not present every configuration option at once.

11 Tables

- Optimise for scanning; show important columns first.
- Avoid unnecessary columns.
- Use row actions rather than adding excessive buttons.
- Use pagination, filtering and search where appropriate.

12 Responsive design

- Design for desktop, tablet and mobile.
- Do not simply shrink desktop layouts.
- Preserve hierarchy and usability across screen sizes.

13 Accessibility

- Maintain sufficient contrast.
- Do not communicate information through colour alone.
- Ensure keyboard accessibility and use semantic HTML.
- Provide accessible labels for icon-only controls.

The design decision rule

“What user problem does this solve?”

Ask it before adding any component. If there is no clear answer, do not add it. If two approaches communicate the same information, choose the simpler one.

UI quality audit

After the UI is generated, instruct your agent to review the actual implementation against the principles, not merely describe what it intended to build.

- Does every major UI element have a clear purpose?
- Are there unnecessary cards, icons, badges, descriptions, borders, shadows or decorative elements?
- Is the visual hierarchy obvious within a few seconds?
- Are spacing, alignment, typography and component treatments consistent?
- Are primary actions clearly distinguishable from secondary ones?
- Could any section be simplified without losing useful information?
- Does the mobile layout preserve hierarchy and usability instead of merely shrinking?
- Are forms easy to scan and complete?
- Are tables optimised for the information users actually need?
- Does the UI remain accessible without relying on colour alone?
- Does the interface feel like one coherent product rather than a collection of generated sections?
- Does the final result feel calm, intentional, professional and easy to learn?

Quality standard

IT SHOULD FEEL	IT SHOULD NOT FEEL
Professional	Over-designed
Calm	Crowded
Intentional	Decorative
Consistent	Childish
Easy to scan	Template-generated
Easy to learn	Full of unnecessary cards, icons, badges or descriptions
Efficient to operate	

Suggested AGENT.md instruction

UI design rule. When creating or refining any UI, always refer to the project's UI/UX design principles before making design decisions. Do not add components for visual decoration alone. After implementation, perform a UI quality audit against the same principles and fix any violations before considering the work complete.

This creates a simple feedback loop: reference → design → implement → audit → refine. The objective is not to make every interface look the same. The objective is to make every design decision intentional.