APPLE DESIGN

دليل استخدام **Skill** التصميم والحركة مع **Claude Code** و**Codex**

من التثبيت إلى **Audit** و**Redesign** لموقع موجود - خطوة بخطوة

نسخة مبسطة وعملية للمطورين ومصممي الواجهات

ما هي Apple Design Skill؟

Apple Design هي Skill مفتوحة المصدر من Emil Kowalski، هدفها تحويل مبادئ Apple في تصميم الواجهات والحركة إلى قواعد عملية يستطيع الـAI Agent تطبيقها على الويب.

هي ليست أداة رسم ولا Template جاهز، وليست Skill رسمية من Apple. هي مجموعة تعليمات تصميم وهندسة حركة مبنية على مبادئ من محاضرات Apple WWDC ومترجمة لتناسب CSS وPointer Events وrequestAnimationFrame ومكتبات الـSpring مثل Motion / Framer Motion.

أفضل استخدام لها: عندما يكون لديك Interface يعمل بالفعل وتريد أن تجعله أكثر استجابة وهدوءًا ووضوحًا وطبيعية في الحركة، أو عندما تبني تفاعلات مثل Bottom Sheets وDrawers وCards وCarousels وSwipe/Drag.

الروابط الرسمية

skills.sh - صفحة الـ**Skill** على https://www.skills.sh/emilkowalski/skills/apple-design

GitHub Repository - https://github.com/emilkowalski/skills

ملف **SKILL.md** مباشرة - https://github.com/emilkowalski/skills/blob/main/skills/apple-design/SKILL.md

ماذا تضيف فعليًا للمشروع؟

- استجابة فورية: Feedback يبدأ مع الضغط أو السحب، وليس بعد انتهاء الحركة.
- **Direct Manipulation**: العنصر يتحرك 1:1 مع حركة المستخدم أثناء Drag / Swipe.
- **Interruptible Motion**: المستخدم يستطيع الإمساك بالعنصر أثناء الحركة وتغيير الاتجاه بدون Jump.
- **Spring Animations**: استخدام Springs بدل Animation ثابتة عندما يكون التفاعل فيزيائيًا.
- **Velocity + Momentum**: استمرار الحركة اعتمادًا على سرعة الإصبع أو الماوس عند الإفلات.
- **Spatial Consistency**: الدخول والخروج من نفس الاتجاه، وربط الـPopover أو Sheet بالمصدر الذي فتحه.
- **Materials & Depth:** Translucency وBlur وDepth بشكل له وظيفة وليس مجرد Glass Effect.

- **Typography:** ضبط Tracking وLeading وOptical sizing بما يناسب حجم النص.
- **Accessibility:** احترام prefers-reduced-motion وreduced-transparency وcontrast.
- **Design Foundations:** Purpose, Agency, Simplicity, Craft, Familiarity وغيرها كقواعد لاتخاذ القرار.

المرحلة 1 - تثبيت الـSkill

أسرع أمر رسمي للتثبيت:

```
npx skills add https://github.com/emilkowalski/skills --skill apple-design
```

الأمر يضيف Skill باسم apple-design، والـCLI يطلب منك اختيار الـAgent المتاح عندك. يمكنك أيضًا تحديد Claude Code أو Codex صراحة حسب بيئتك.

Prompt جاهز للتثبيت - Claude Code / Codex

```
Open the official Apple Design skill source first:
https://github.com/emilkowalski/skills/tree/main/skills/apple-design

Then install ONLY the apple-design skill from this repository using the
official skills CLI.
Use this command unless my environment requires a documented equivalent:
npx skills add https://github.com/emilkowalski/skills --skill apple-design

Before changing anything:
1. Detect whether I am using Claude Code, Codex, or another supported agent.
2. Tell me whether you will install it for this project only or globally.
3. Do not overwrite unrelated skills or agent configuration.
4. Install apple-design only.
5. Verify that the skill is discoverable after installation.
6. Do not modify my application code yet.
7. Report the exact install location and how I can remove/update the skill
later.

After verification, stop and tell me that the skill is ready.
```

مهم قبل التثبيت

- اشتغل داخل Git branch منفصل لو المشروع مهم.
- اقرأ SKILL.md قبل تشغيلها على مشروع Production.
- الـSkill تعليمات للـAgent؛ لا تعطيها أسرار أو صلاحيات لا يحتاجها المشروع.

• التثبيت لا يعني أن شكل موقعك سيتغير وحده. يجب أن تطلب Audit أو Redesign بشكل واضح.

اختبار سريع بعد التثبيت

```
Use the apple-design skill to review one existing interactive component in
this project. Do not edit code. Tell me what violates the skill's principles
around response, interruptibility, springs, velocity, spatial consistency,
typography, and accessibility. Return a short prioritized audit only.
```

المرحلة 2 - عندك موقع موجود؟ ابدأ بـ Audit قبل الـRedesign

لا تبدأ بطلب: "خلي الموقع Apple". الأفضل أن تجعل الـAgent يفهم المشروع أولًا، ثم يراجع ما يحتاج تعديلًا، ثم يعطيك خطة واضحة.

Prompt - Audit فقط بدون تعديل

```
Use the installed apple-design skill to audit my existing website before
changing any code.

First, inspect the current frontend architecture, routes, global styles,
design tokens, components, motion/animation libraries, responsive behavior,
and accessibility patterns.

Then review the UI against the apple-design principles, especially:
- immediate feedback and perceived latency
- direct manipulation for drag/swipe interactions
- interruptible motion
- springs vs fixed-duration transitions
- velocity handoff and momentum where relevant
- spatial consistency and transform origins
- translucent materials and depth
- typography: hierarchy, tracking, leading, optical sizing
- reduced motion / transparency / contrast accessibility
- simplicity, familiarity, agency, and craft

Do NOT redesign or edit files yet.
Return:
1. What is already good and should be preserved.
2. The top UI/UX problems ranked High / Medium / Low impact.
3. Exact components/files likely involved.
4. A proposed redesign direction that keeps my brand identity.
5. A phased implementation plan.
6. Risks: performance, accessibility, bundle size, or browser support.

Avoid making the site look like a fake Apple clone. Apply the principles, not
Apple's branding.
```

ماذا تنتظر من الـAudit؟

• قائمة بالمشكلات بدل تعديلات عشوائية.

• تحديد المكونات التي تحتاج Motion فعليًا والتي يجب أن تبقى بسيطة.

• الحفاظ على الهوية الحالية بدل نسخ Apple بصريًا.

• اكتشاف مشاكل Accessibility وMotion قبل إضافة Effects جديدة.

• خطة تنفيذ تستطيع مراجعتها قبل أن يلمس الـAgent الكود.

المرحلة 3 - Prompt تنفيذ Redesign لموقع موجود

```
Use the installed apple-design skill and the audit you just completed to
improve this existing website.

Goal: make the interface feel more polished, responsive, physical, and premium
while preserving the product's own brand identity.

Rules:
- Preserve all existing business logic, API integrations, routes, forms,
authentication, SEO behavior, and data flow unless a change is technically
required.
- Do not turn the site into an Apple visual clone.
- Keep the current content and information architecture unless the audit
proves a change improves usability.
- Reuse the existing design system/tokens where possible; improve them instead
of creating random new values.
- Prefer subtle motion. Motion must communicate state, hierarchy, direction,
or causality.
- Use springs for gesture-driven components where appropriate.
- Make interactions interruptible and start from the current visual state.
- For drag/swipe, use direct 1:1 tracking and preserve release velocity when
useful.
- Keep enter/exit paths spatially consistent.
- Improve typography hierarchy, spacing, tracking, and line-height.
- Use translucency/depth only where it communicates hierarchy.
- Implement prefers-reduced-motion and accessible fallbacks.
- Keep animations performant: prefer transform/opacity and avoid unnecessary
layout work.
- Keep the result responsive on mobile, tablet, and desktop.

Implementation process:
1. Create a small plan and list the files you will change.
2. Implement in focused batches, starting with global foundations and the
highest-impact components.
3. After each batch, run the relevant lint/typecheck/tests/build.
4. Check responsive states and keyboard accessibility.
```

```
5. Summarize what changed, why, and any remaining improvements.

Start with the highest-impact improvements from the audit. Do not rewrite the
whole project unnecessarily.
```

Prompt مختصر لو عايز تحسين Component واحد

```
Use apple-design to improve this component only: [COMPONENT/PATH]. Preserve
its API and behavior. Make the interaction feel immediate, interruptible,
spatially consistent, and accessible. Use spring/velocity principles only
where the interaction is physical. Explain the changes briefly, then implement
and test them.
```

Prompts عملية جاهزة

Bottom Sheet / Drawer

```
Review and rebuild this bottom sheet using apple-design. It must track the
pointer 1:1 while dragging, preserve the grab offset, hand off release
velocity to a spring, choose the final snap point using momentum, and remain
fully interruptible while closing/opening. Add reduced-motion behavior.
```

Cards / Press Interaction

```
Use apple-design to make this card interaction feel physical without becoming
flashy. Add immediate pointer-down feedback, subtle spring recovery,
keyboard/focus parity, and reduced-motion fallback. Keep the card layout and
business behavior unchanged.
```

Navbar / Translucent Material

```
Use apple-design to review this sticky navigation. Improve hierarchy and depth
with a restrained translucent material only if it helps legibility. Add a
reduced-transparency fallback and avoid stacking glass-on-glass surfaces.
```

Typography

```
Audit this page typography using apple-design. Improve hierarchy, tracking,
leading, responsive type sizing, and readability. Keep the brand font if it
works; do not replace it just to imitate Apple.
```

Carousel / Swipe

```
Use apple-design to improve this carousel. Track drag 1:1, preserve release
velocity, project momentum to choose the target slide, add soft edge
resistance, and make the animation interruptible. Preserve keyboard controls
and reduced-motion support.
```

Full UI Polish

```
Use apple-design as a review lens for this page. Identify the 5 highest-impact
polish improvements across feedback, motion, spacing, typography, depth, and
accessibility. Implement only those 5 changes and avoid decorative motion that
does not improve understanding.
```

أفضل Workflow للاستخدام

1	ثبّت الـSkill وتأكد أنها ظاهرة للـAgent.
2	اعمل Audit بدون تعديل كود.
3	حدد أعلى 3-5 مشاكل لها تأثير حقيقي.
4	نفذ التعديلات في Branch منفصل وبـBatches صغيرة.
5	اختبر Mobile/Desktop + Keyboard + Reduced Motion.
6	راجع الأداء: لا تضف Animation على كل شيء.
7	اعمل Build/Lint/Typecheck واختبر الـFlows المهمة.
8	قارن Before / After وحدد هل التجربة أصبحت أوضح فعلًا أم فقط أكثر حركة.

نقاط مهمة حتى لا يكون التصميم AI-looking

• لا تطلب "Apple style" وحدها. اشرح وظيفة الصفحة والهدف من الحركة.

• لا تستخدم Glass وBlur في كل Card. استخدم الـMaterial فقط عندما يوضح Hierarchy.

• الـSpring ليست مناسبة لكل Animation. استخدمها عندما يوجد تفاعل فيزيائي أو عنصر يمكن لمسه/سحبه.

• الحركة يجب أن تكون قابلة للإيقاف أو التغيير إذا كان المستخدم يتفاعل معها.

• الأولوية دائمًا للوضوح، السرعة، الاستجابة، والـAccessibility.

• حافظ على Brand Colors والخطوط والهوية؛ طبق مبادئ الحركة وليس شكل Apple حرفيًا.

Checklist قبل تسليم الـRedesign

☐ الـBusiness logic لم يتغير

☐ الـAPI والـForms تعمل

☐ Responsive سليم

☐ Keyboard navigation يعمل

☐ Focus states واضحة

☐ Reduced Motion موجود

☐ Animation لا يوجد تسبب Lag

☐ Enter/Exit paths منطقية

☐ Typography واضحة

☐ الـUI ما زال يحمل هوية المنتج وليس نسخة Apple

الخلاصة

Apple Design Skill مفيدة عندما تستخدمها كـDesign Review System وليس كفلتر شكلي. أقوى نتيجة تأتي من:

Audit → اختيار التحسينات المهمة → تنفيذ تدريجي → اختبار الحركة والأداء والـAccessibility.

المصادر

https://www.skills.sh/emilkowalski/skills/apple-design

https://github.com/emilkowalski/skills

https://github.com/emilkowalski/skills/blob/main/skills/apple-design/SKILL.md