

Construire une LP animée en 12 minutes avec Stitch et Claude Code

Stitch (Google) génère le design depuis une description en langage naturel. Claude Code transforme ces écrans en site Next.js production-ready avec animations Framer Motion. Guide complet : prompts prêts à copier, étapes d'export MCP, configuration Claude Code, déploiement Vercel, débogage des cas tordus.

GUIDE · 28 elements · 12 pages

Coder une landing page à la main, c'est 1 à 3 jours par itération si tu pars de zéro. Pour un freelance, un fondateur, un créateur qui doit lancer quelque chose vite, c'est un goulot d'étranglement absurde en 2026. Le design existe dans ta tête en 5 minutes, et tu passes 20 heures à l'écrire en HTML/CSS.

Ce guide documente le workflow que j'utilise pour passer d'un brief texte à un site Next.js animé, déployé sur Vercel, en moins de 15 minutes. Deux outils, tous les deux gratuits ou quasi : Stitch (Google) pour la partie design, Claude Code (Anthropic) pour le code. La passerelle entre les deux passe par MCP, le protocole d'interopérabilité qui permet à Claude Code de lire les écrans Stitch directement.

Compter 10 minutes de setup la première fois (installation Claude Code, comptes Google et Vercel). Ensuite, chaque nouvelle LP prend entre 12 et 25 minutes selon la complexité. Le guide est rangé en 14 étapes opérationnelles, plus les prompts prêts à copier en bas, plus une liste d'erreurs courantes.

ÉTAPE 01

prérequis. Tu as un compte Google (pour Stitch), Claude Code installé sur ta machine (commande : `npm install -g @anthropic-ai/claude-code`, ou via `brew install anthropic/claude/claude-code` sur Mac), un compte Vercel (gratuit jusqu'à 100 GB de bande passante). Compter 10 minutes de setup si rien n'est en place.

ÉTAPE 02

ouvrir Stitch. Aller sur <https://stitch.withgoogle.com> et se connecter avec ton compte Google. Choisir Web comme cible (pas App, sauf si tu veux générer une interface mobile native). Le mode par défaut est suffisant pour une LP. Ignorer pour l'instant le bouton DESIGN.md (utile pour la cohérence multi-pages, on y vient plus tard).

ÉTAPE 03

prompter la LP. Le prompt Stitch idéal contient 7 ingrédients : type de page + nom du business, secteur, direction artistique (style), nombre et nature des sections, hero principal, design system, librairie d'animation. Voir le template ci-dessous. Stitch génère mieux en anglais qu'en français : les outputs sont 30 pour cent plus riches visuellement. Prompt court (1 paragraphe), pas un cahier des charges de 2 pages.

ÉTAPE 04

itérer le design. Stitch propose souvent 2 ou 3 variantes. Choisir la plus proche, puis utiliser le chat intégré pour ajuster : 'darker palette', 'more whitespace', 'add a pricing section between testimonials and FAQ', 'change hero illustration to abstract gradient'. Faire 3 à 5 itérations max. Au-delà, soit le prompt initial est mauvais, soit tu cherches une perfection qui freine la suite.

ÉTAPE 05

export MCP. Quand le design est validé, cliquer Export en haut à droite. Choisir MCP (pas Figma, pas .zip). Stitch ouvre une modale avec une commande à copier. Cette commande contient un token

unique d'accès à ton projet Stitch. Garder la modale ouverte.

ÉTAPE 06

configuration MCP côté Claude Code. Ouvrir un terminal dans un dossier vide (ex : ~/projects/ma-lp). Lancer la commande copiée depuis Stitch : elle configure le serveur MCP Stitch dans ton fichier .claude/mcp.json. Vérifier la connexion : `claude mcp list`. Stitch doit apparaître avec statut `connected`. Si erreur d'auth, régénérer la commande depuis Stitch.

ÉTAPE 07

lancer Claude Code. Toujours dans le même dossier, taper `claude`. Attendre que la session s'initialise. Claude Code détecte automatiquement le serveur MCP Stitch et a accès à tes écrans. Confirmer avec : 'Liste les écrans Stitch disponibles'. Tu dois voir tes pages.

ÉTAPE 08

prompt principal Claude Code. Coller le prompt complet (voir ci-dessous). Il dit à Claude de scaffolder un projet Next.js 15, d'importer le design system Stitch, de coder toutes les sections avec animations Framer Motion, de connecter les écrans, et de préparer un déploiement Vercel. Compter 5 à 8 minutes de génération. Pendant ce temps, ne pas toucher au terminal.

ÉTAPE 09

connecter les écrans. Stitch te sort plusieurs écrans (souvent : home, à-propos, pricing, contact). Claude Code les fusionne en une seule page scrollable avec ancrs de navigation, ou en plusieurs routes si tu as demandé du multipage. Vérifier le routing : `npm run dev`, ouvrir `localhost:3000`, scroller toutes les sections, cliquer chaque lien d'ancre.

ÉTAPE 10

animations. Les entrance animations (fade-in-up au scroll) sont gérées par Framer Motion via `whileInView` avec `viewport once:true`. Les micro-interactions (hover sur les cartes, magnetic CTA buttons, smooth scroll) sont codées en CSS + framer-motion. Si une animation est cassée ou absente, demander à Claude : 'Ajoute une animation parallax sur la hero section, le titre monte plus lentement que le background'.

ÉTAPE 11

preview local. Lancer `npm run dev`. Ouvrir `localhost:3000` dans Chrome. Tester sur 3 viewports : desktop (1440px), tablette (768px), mobile (390px iPhone). Chrome DevTools > Toggle device toolbar (Cmd+Shift+M). Noter les bugs : sections qui débordent en mobile, textes trop gros, espacement cassé.

ÉTAPE 12

fix responsive. Pour chaque bug, retourner dans Claude Code et formuler la correction précisément : 'La section pricing en mobile a les 3 cartes empilées correctement mais le titre de la carte du milieu déborde, réduis `text-2xl` à `text-xl` en dessous de md'. Claude patche immédiatement. Re-tester. Compter 3 à 5 corrections en moyenne.

ÉTAPE 13

déploiement Vercel. Dans le terminal du projet, taper `vercel`. Première fois, il te demande de te connecter et de lier le projet. Confirmer les paramètres par défaut : framework Next.js détecté automatiquement, root directory `.`, build command `next build`, output directory `.next`. Pousser : `vercel --prod`. Vercel renvoie l'URL live en 30 à 60 secondes.

ÉTAPE 14

domaine custom (optionnel). Dans le dashboard Vercel, Project > Settings > Domains, ajouter ton domaine custom (ex : `lp.tondomaine.com`). Vercel te donne 2 entrées DNS à ajouter chez ton registrar (Cloudflare, Gandi, OVH). Propagation 5 à 60 minutes. Une fois propagé, le site est accessible sur ton domaine, SSL géré automatiquement.

BONUS A

prompt Stitch template (à remplir). Design a cinematic landing page for "[NOM]", a [SECTEUR]

business. Visual direction: [STYLE descriptors]. Fully scrollable with [N] sections: [section 1], [section 2], ..., [section N]. Hero with [TYPE D'ANIMATION]. Apply a [DESIGN SYSTEM] with [PALETTE] and [TYPO]. Use Framer Motion for entrance animations and micro-interactions.

BONUS B

exemple SaaS premium (à copier dans Stitch). Design a cinematic landing page for "Arc", a stealth-mode B2B AI orchestration platform. Visual direction: editorial dark mode with subtle aurora gradients, generous whitespace, serif display headlines paired with Inter body. Fully scrollable with 7 sections: hero with animated text reveal and gradient mesh, feature grid with hover micro-interactions, product dashboard mockup with parallax, integrations marquee, pricing tiers with sticky highlight, social proof with company logo grid, footer with sticky CTA. Apply a liquid-glass design system with deep navy base, violet accents, soft glow shadows. Use Framer Motion for staggered entrance animations and scroll-triggered reveals.

BONUS C

exemple cours en ligne (à copier dans Stitch). Design a cinematic landing page for "Vertex Academy", an online program teaching quantitative trading. Visual direction: financial editorial with subtle grid background, terminal-inspired UI details, deep navy with electric green accents. Fully scrollable with 8 sections: hero with floating price tickers and animated text reveal, curriculum overview with module cards, instructor section with portrait and credentials, dashboard preview with chart animations, testimonial wall, pricing with three tiers, FAQ accordion, footer with secondary CTA. Apply a Bloomberg-terminal-inspired design system with mono details. Use Framer Motion for entrance animations and chart-style micro-animations on hover.

BONUS D

prompt Claude Code complet (à coller après l'import MCP). You now have access to my Stitch design via MCP. Build the complete production-ready Next.js 15 site from these screens. 1. Scaffold a Next.js 15 app with TypeScript, Tailwind CSS v4, and Framer Motion. 2. Connect all screens from the Stitch design into a single scrollable page with anchor navigation in the header. 3. Faithfully implement the design system: pull color palette, typography, spacing, border radii and shadows from Stitch, set them as Tailwind theme tokens in globals.css. 4. Implement every section with semantic HTML and accessible markup. Use server components by default, client components only when needed. 5. Add Framer Motion entrance animations on each section: fade-in-up with stagger, using `whileInView` with `viewport once:true` and `amount:0.2`. 6. Add micro-interactions: hover lift on cards, magnetic CTA buttons, smooth scroll on anchor links. 7. Make the layout fully responsive: mobile-first, breakpoints at sm/md/lg/xl, all sections must look great on 390px iPhone viewport. 8. Optimize performance: next/image for all images, dynamic imports for heavy components, lazy-load below-the-fold. 9. SEO basics: metadata in app/layout.tsx, OpenGraph image, Organization and WebSite structured data. 10. Init git repo, commit with conventional commits, prepare Vercel deployment via vercel.json. Output the file tree first, then build each file in order. Ask before installing any package not in spec.

ERREUR 01

Stitch génère un design fade ou générique. Cause probable : prompt trop court, manque de descripteurs visuels précis. Fix : rajouter 3 à 5 adjectifs de direction artistique (editorial, brutalist, glassmorphism, cinematic, minimal, maximalist). Préciser la palette en mots concrets (deep navy with electric green, monochrome with chromatic accents).

ERREUR 02

Claude Code dit que le serveur MCP Stitch n'est pas accessible. Cause : le token MCP a expiré ou n'a pas été correctement injecté. Fix : retourner dans Stitch, cliquer Export, MCP, copier la nouvelle commande, l'exécuter dans le terminal. Vérifier `claude mcp list` après.

ERREUR 03

les animations Framer Motion ne se déclenchent pas. Cause fréquente : viewport once:true mais le composant n'a pas atteint le seuil amount au premier rendu. Fix : baisser amount à 0.1 ou 0.05, ou utiliser margin:"-100px" pour anticiper le déclenchement.

ERREUR 04

la version mobile a des sections qui débordent à droite. Cause : un élément avec width fixe en pixels ou un grid mal configuré. Fix : remplacer width: 1200px par max-width: 1200px + width: 100%. Vérifier les grilles : grid-cols-1 md:grid-cols-2 lg:grid-cols-3 pour les changements de breakpoint propres.

ERREUR 05

Vercel échoue au build avec une erreur de types TypeScript. Cause : Claude Code a généré du code rapide avec quelques any ou des props mal typées. Fix : relire l'output, identifier les any, demander à Claude : 'Type strictement tous les composants, élimine tous les any, ajoute les interfaces manquantes'.

ERREUR 06

les images sont énormes et le LCP est mauvais (Lighthouse score sous 60). Cause : images PNG/JPG en taille originale. Fix : convertir tout en WebP. Dans Claude Code : 'Remplace toutes les images par next/image, génère des placeholders blur, ajoute sizes correct pour chaque image en fonction de son rôle (hero, card, thumbnail)'.

ERREUR 07

la typographie ne charge pas, le texte clignote au premier render (FOIT/FOUT). Cause : @font-face mal configuré ou next/font absent. Fix : utiliser next/font/google avec display:'swap' et fallback fonts dans globals.css. Pour Inter : import { Inter } from 'next/font/google'.

ERREUR 08

le scroll smooth ne fonctionne pas sur les anchor links. Cause : html { scroll-behavior: smooth } absent ou Framer Motion intercepte. Fix : dans globals.css, ajouter html { scroll-behavior: smooth }. Pour les liens, utiliser <Link href="#section" scroll={true}>.

ERREUR 09

Vercel deploy réussi mais le site est lent au premier load. Cause : trop de client components, pas assez de server components. Fix : auditer chaque composant. Tout ce qui n'a pas d'interactivité (useState, useEffect, onClick) doit être server component. Seuls les composants interactifs portent 'use client'.

ERREUR 10

tu veux refaire 5 LP de variantes mais Stitch génère trop différemment à chaque fois. Cause : pas de DESIGN.md partagé entre les sessions. Fix : créer un DESIGN.md à la fin de ta première LP réussie (Stitch le génère automatiquement à l'export), le réutiliser comme input pour toutes les LP suivantes. Cohérence garantie sur la palette, la typo, les espacements.

POUR FINIR

Ce workflow ne remplace pas un vrai design system tenu sur la durée pour un produit qui scale. Pour ça, tu as toujours besoin d'un designer humain qui pense l'identité, la cohérence cross-pages, l'évolution dans le temps. Stitch et Claude Code sont fantastiques pour la phase 0 (lancement, test, MVP) et pour les LP one-shot (page de lancement produit, page d'événement, portfolio).

Cas d'usage idéaux que j'ai validés : page de lancement produit, page d'événement (webinaire, conférence), portfolio personnel, MVP de produit SaaS pour valider une idée, page promo limitée dans le temps. Pour chacun, tu peux générer 5 à 10 variantes en une matinée et tester réellement, au lieu de discuter design en réunion pendant trois semaines.

Itère vite. La force du stack n'est pas dans une LP parfaite, elle est dans la capacité à en produire 20 et garder celle qui convertit. Ne tombe pas amoureux du premier output Stitch. Génère, déploie, mesure,

jette, recommence.

