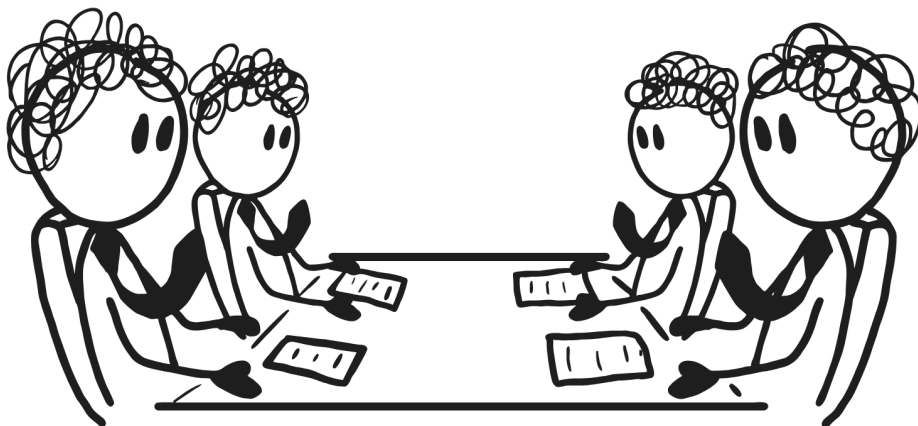


Design Premium Websites with AI

The Complete Follow-Along Guide

Transform AI from a guessing tool into your design partner



Introduction

75% of users won't trust a company if the design doesn't look good. This guide will show you how to get premium, gorgeous designs from AI that would typically cost \$5,000+ from a design agency.

The secret? **Give AI the same context you'd give a real designer.**

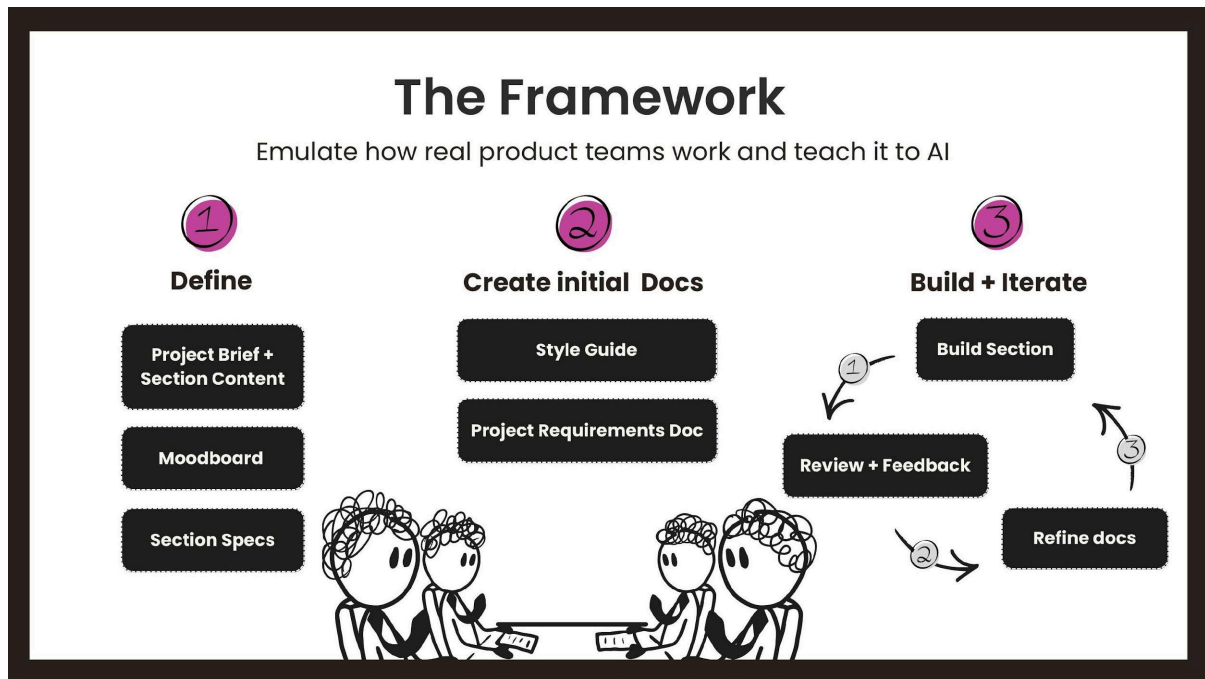
Why This Framework Works

- Mirrors how real product teams work — each section has clear scope and single source of truth
- Gives AI clean, isolated context — no pollution from unrelated sections
- Makes the process iterative and scalable — refine one section without breaking others

💡 *This is specs-driven development applied to AI: clear specs first, build second, then iterate.*

The Framework Overview

The framework consists of three main phases:

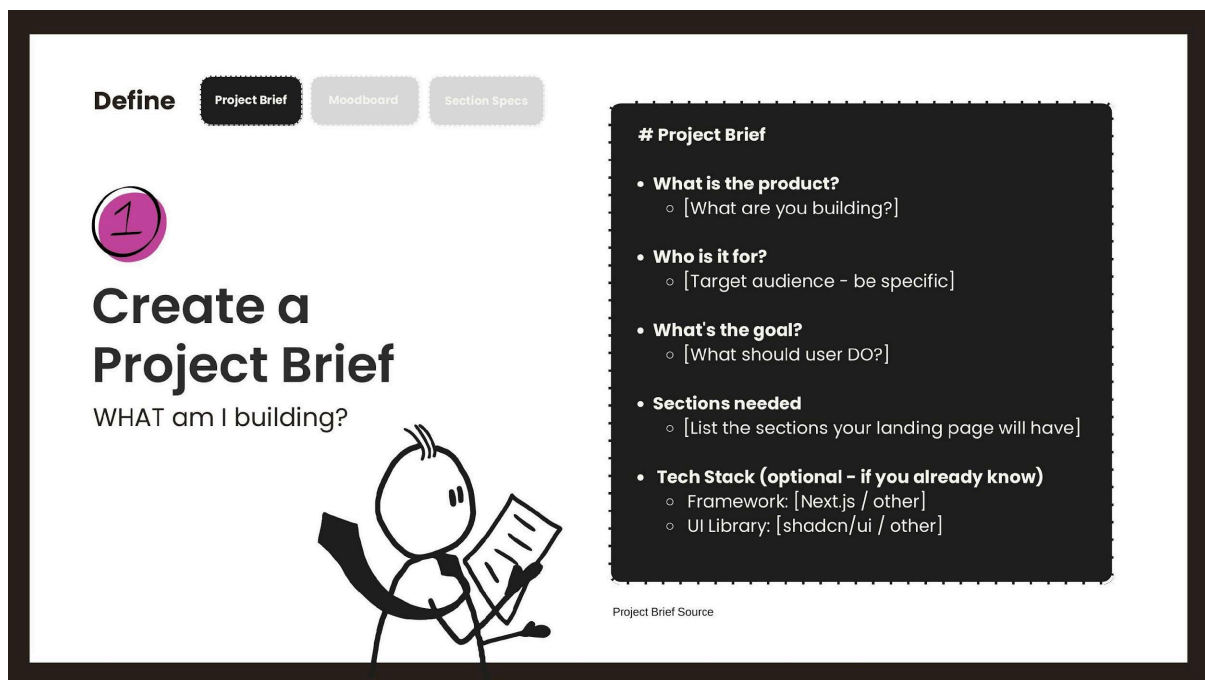


Phase 1: Define (Your Preparation)

Before talking to AI, you need to prepare these documents. Think of AI as a new team member — you wouldn't tell them "build me a website" without any context.

Step 1: Create Your Project Brief

The project brief explains WHAT you're building and WHY. This is your foundation document.



Template: project-brief.md

```
# Project Brief

## What is it?
[One paragraph describing the product]

## Who is it for?
[Target audience - be specific]

## What's the goal?
[What action should visitors take?]

## Sections needed
1. Navigation Bar
2. Hero Section
3. Trust Logos
4. Features
5. How It Works
6. Pricing
7. Testimonials
8. FAQ
9. Footer
```

```
## Additional Requirements
- Fully responsive
- Fast loading
- [Any other requirements]
```

Adding Project Brief to Cursor

Once you've created your project brief, add it to Cursor (or your AI tool):

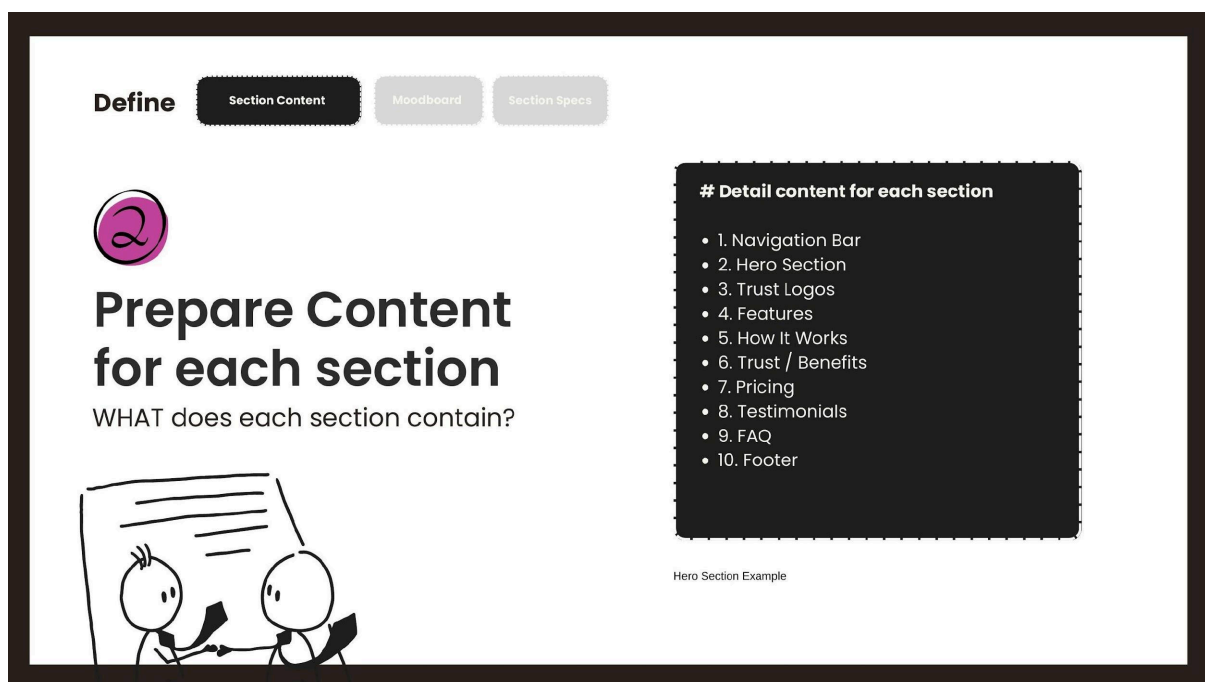
- Create a folder called website-guidelines in your project root
- Create a markdown file called product-brief.md inside this folder
- Paste your project brief content into this file

💡 *Keep your project brief in a Google Doc first for easy editing, then copy to Cursor when ready.*

```
your-project/
├── website-guidelines/
│   └── product-brief.md    ← Your project brief goes here
```

Step 2: Prepare Content for Each Section

For each section listed in your project brief, create a separate markdown file with all the content.



Template: [section-name]-content.md

```
# [Section Name] Content

## Headlines
- Main: "Your exact headline text"
- Sub: "Your description text"
```

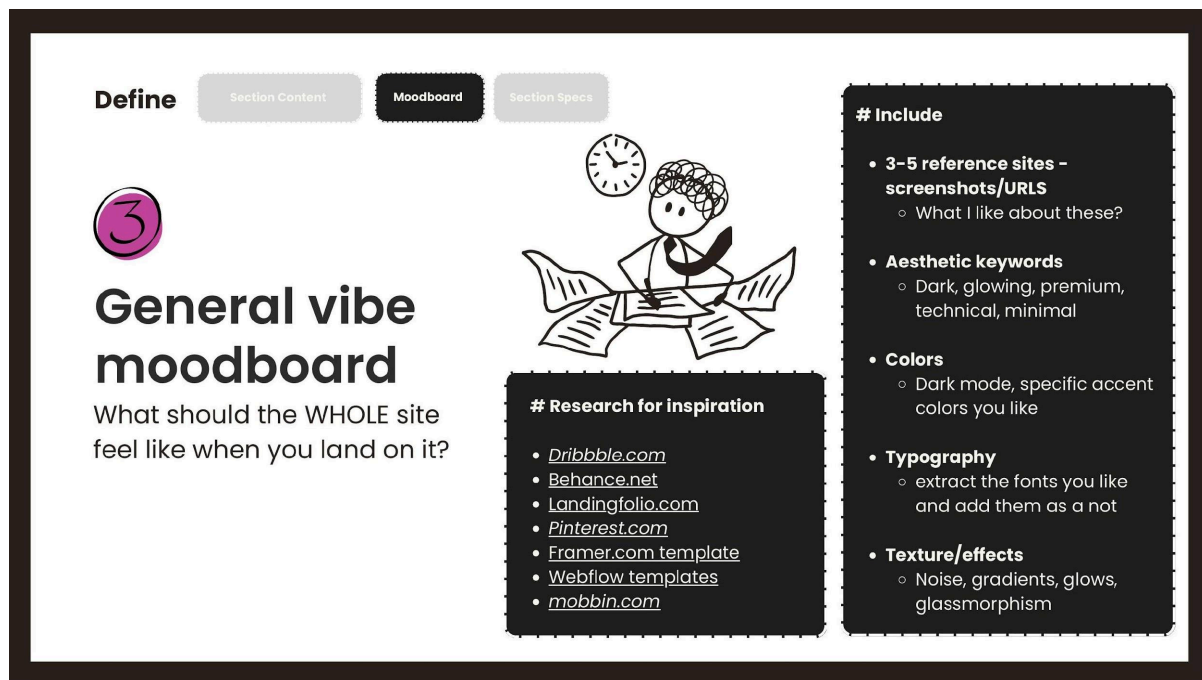
```
## Buttons / CTAs
- Primary: "Button label"
- Secondary: "Button label"

## Body Text
[Any paragraph content for this section]
```

💡 *Create a separate content file for EACH section. You'll combine this with design specs later.*

Step 3: Create Your General Vibe Moodboard

This answers: "What should the WHOLE site feel like when you land on it?"



Where to Research Inspiration

- [Dribbble.com](https://dribbble.com) — Design inspiration
- [Behance.net](https://behance.net) — Portfolio showcases
- [Landingfolio.com](https://landingfolio.com) — Landing page examples
- [Pinterest.com](https://pinterest.com) — Visual mood boards
- [Framer.com/templates](https://framer.com/templates) — Modern templates
- [Webflow templates](https://webflow.com/templates) — Professional templates
- [Mobbin.com](https://mobbin.com) — Mobile & web patterns

Template: 0.design-moodboard.md

```
# General Design Moodboard

## Inspiration References
### Reference 1: [Site Name]
- URL: [link]
- Screenshot: [attach image]
- What I like: [specific elements]

## Aesthetic Keywords
[Dark, glowing, premium, technical, minimal, etc.]

## Colors
- Theme: [Dark mode / Light mode]
- Primary: [#hexcode] - [why this color]
- Secondary: [#hexcode]
- Background: [#hexcode]
```

```
## Typography
- Headlines: [Font name]
- Body: [Font name]
- Accent: [Font name for special elements]

## Effects & Textures
[Noise, gradients, glows, glassmorphism, etc.]
```

Real Example: Generic Vibe Moodboard

Here's a real example of a moodboard for a security SaaS landing page:



Notice how specific the notes are:


- Main Color: "I like the dark theme → perfect for tech/dev audience"
- Color choice explained: "#01FFFF evokes trust and technical vibe"
- Typography specified: "Satoshi + Instrument Serif Regular Italic"

- Effects detailed: "Dark elegant background with subtle noise and glowing particles"
- Complementary color: "Orange #ffa260 only for glow, highlights, subtle accents — NOT for text"

Adding Moodboard to Cursor

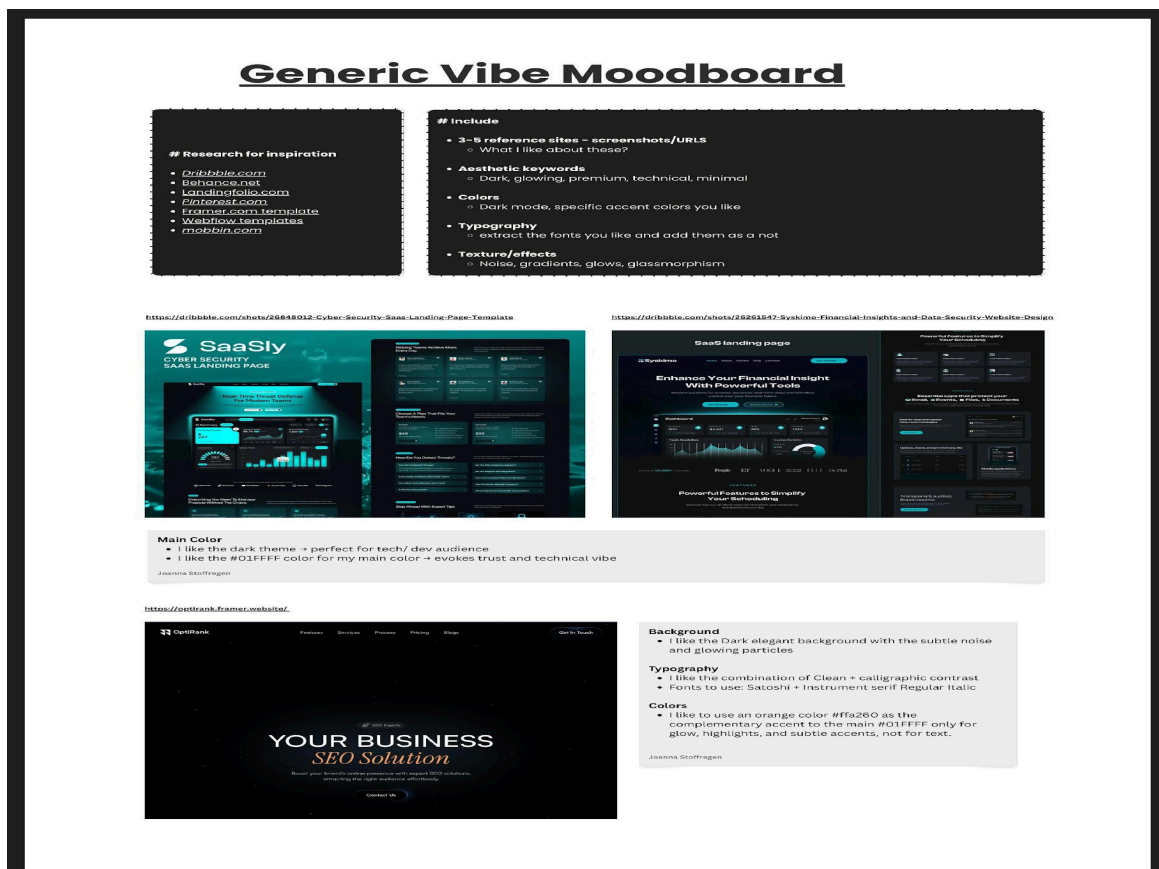
Create a folder called website-sections and add your moodboard as a markdown file:

```
your-project/
├── website-guidelines/
│   └── product-brief.md
├── website-sections/
│   ├── inspo-images/           ← Your screenshots go here
│   └── 0.design-moodboard.md   ← General vibe moodboard
```

 *Export your moodboard images and reference them in the markdown file. AI can see images when you include them!*

Real Example: Generic Vibe Moodboard

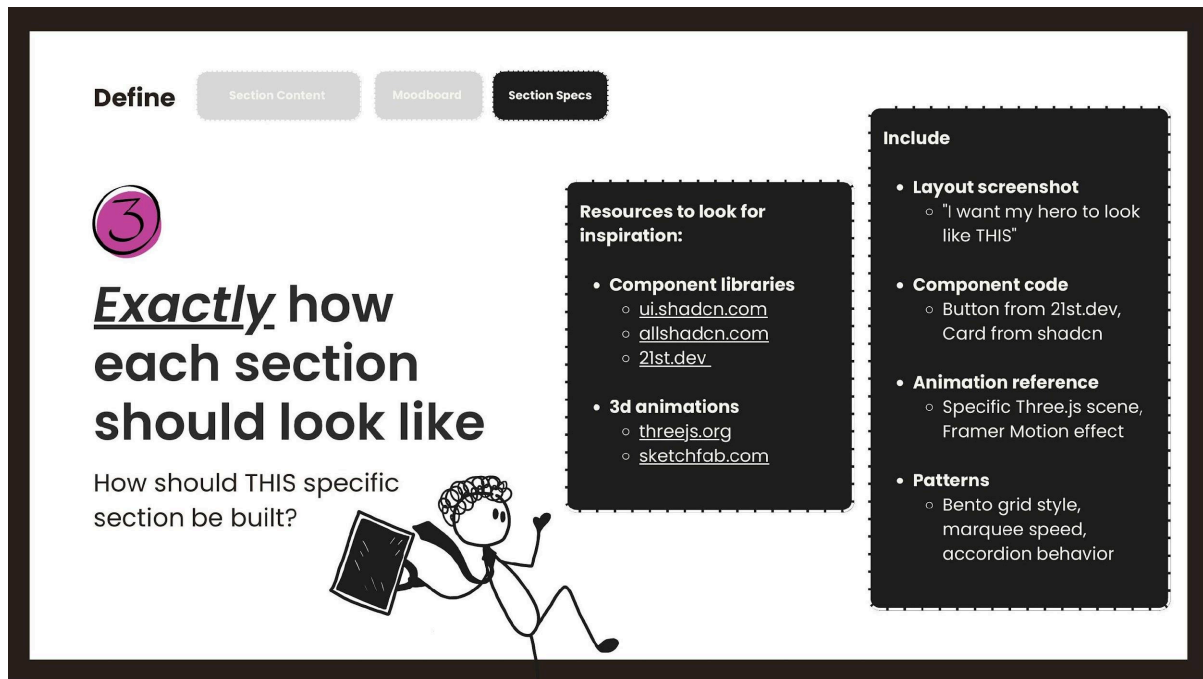
Here's a real moodboard from the Guardian AI project showing exactly how detailed your notes should be:



Notice how the moodboard includes: reference URLs, specific color codes (#01FFFF), font names (Satoshi + Instrument Serif), and notes explaining WHY each choice was made.

Step 4: Create Section-Specific Specs

This is where you get VERY detailed about exactly how each section should look. Think of it like giving instructions to a designer.



Resources for Components & Animations

Component Libraries:

- ui.shadcn.com — Base UI components
- 21st.dev — Premium components built on shadcn/ui (includes code!)
- allshadcn.com — Extended shadcn components

3D Animations:

- threejs.org — 3D graphics library with examples
- sketchfab.com — 3D model marketplace

How to Use 21st.dev Components

When you find a component you like on 21st.dev:

- Click on the component to see the full code
- Copy the component code and paste it into your section spec file
- Also copy the "prompt" they provide — it includes installation instructions for dependencies
- AI will use this code as a reference and install the required libraries automatically

💡 21st.dev components often include prompts with dependency installation instructions. Always copy these too!

Real Example: Section-Specific Moodboard

Here's the actual section-specific moodboard from the Guardian AI project. Notice how DETAILED each section is

Section Specific Moodboard

Resources to look for inspiration:

- Component libraries**
 - [shadcn/ui](#)
 - [shadcn/ui](#)
 - [ZuiLab](#)
- Animations**
 - [Motion](#)
 - [shadcn/ui](#)

Include:

- Layout screenshot**
 - "I want my hero to look like this"
- Component code**
 - Button from the dev, Card from shadcn
- Animation reference**
 - Specific Three.js scene, frame Motion effect
- Patterns**
 - Bento grid style, marquee speed, accordion behavior

Hero Section

Reference (Bento)

- [https://twitter.com/robertmiller](#)

Design Notes:

- Use the 3D model as the main hero element, but the glowing shape is the focus.
- Use the 3D model as the main hero element, but the glowing shape is the focus.
- Use the 3D model as the main hero element, but the glowing shape is the focus.

Hero Section:

- Use the 3D model as the main hero element, but the glowing shape is the focus.
- Use the 3D model as the main hero element, but the glowing shape is the focus.
- Use the 3D model as the main hero element, but the glowing shape is the focus.

Trust Icons

Trust Icons Section:

- Use the logos as the main hero element, but the glowing shape is the focus.
- Use the logos as the main hero element, but the glowing shape is the focus.
- Use the logos as the main hero element, but the glowing shape is the focus.

Features

Features Section:

- Use the icons as the main hero element, but the glowing shape is the focus.
- Use the icons as the main hero element, but the glowing shape is the focus.
- Use the icons as the main hero element, but the glowing shape is the focus.

How it works

How it works:

- Use the diagram as the main hero element, but the glowing shape is the focus.
- Use the diagram as the main hero element, but the glowing shape is the focus.
- Use the diagram as the main hero element, but the glowing shape is the focus.

Core Capabilities

Core Capabilities Section:

- Use the icons as the main hero element, but the glowing shape is the focus.
- Use the icons as the main hero element, but the glowing shape is the focus.
- Use the icons as the main hero element, but the glowing shape is the focus.

Pricing Section

Pricing Section:

- Use the table as the main hero element, but the glowing shape is the focus.
- Use the table as the main hero element, but the glowing shape is the focus.
- Use the table as the main hero element, but the glowing shape is the focus.

Testimonial

Testimonial Section:

- Use the quote as the main hero element, but the glowing shape is the focus.
- Use the quote as the main hero element, but the glowing shape is the focus.
- Use the quote as the main hero element, but the glowing shape is the focus.

FAQ Section

FAQ Section:

- Use the list as the main hero element, but the glowing shape is the focus.
- Use the list as the main hero element, but the glowing shape is the focus.
- Use the list as the main hero element, but the glowing shape is the focus.

Each section includes: reference screenshots, component URLs from 21st.dev, specific code to use, effect descriptions, and layout instructions.

Template: [X].[section-name].md

```
# [Section Name] Specifications


## Content
[Copy from your section content file]

## Layout Reference
- Screenshot: [attach reference image]
- Source URL: [where you found it]

## Components to Use
### [Component Name]
- Source: [21st.dev / shadcn / custom]
- Description: [pill-shaped, dark bg, glow outline]
- Code:
  ```tsx
 [Paste the component code here]
  ```

## Animations / Effects
### [Animation Name]
- Type: [Three.js / Framer Motion / CSS]
- Reference: [URL or description]
- Code:
  ```tsx
 [Paste animation code]
  ```

## Design Instructions
[Any additional notes about spacing, behavior, etc.]
```

 *Pro Tip: When you find a component on 21st.dev, they often include installation prompts. Copy these into your section specs so AI knows how to install dependencies!*

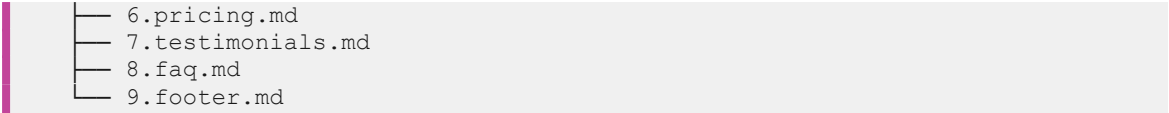
Adding All Sections to Cursor

Now that you have your moodboard and section specs ready, add them to Cursor. This folder structure is critical for AI to have clean, isolated context.

Create the website-sections folder

In your project root, create a folder called website-sections with this structure:

```
your-project/
├── website-guidelines/
│   └── product-brief.md
├── website-sections/
│   ├── inspo-images/           ← Your reference screenshots
│   ├── 0.design-moodboard.md   ← General vibe moodboard
│   ├── 1.navigation-bar.md     ← Section spec + content
│   ├── 2.hero-section.md
│   ├── 3.trust-logos.md
│   ├── 4.features.md
│   └── 5.how-it-works.md
```



```
6.pricing.md
7.testimonials.md
8.faq.md
9.footer.md
```

What each section file should contain

Each section file (1.navigation-bar.md, 2.hero-section.md, etc.) should include BOTH:

- The CONTENT for that section (headlines, buttons, body text)
- The DESIGN SPECS for that section (layout, components, animations, code)

💡 *Be VERY detailed. Include reference images, component code from 21st.dev, Three.js animation code, specific colors, fonts — everything AI needs to build exactly what you want.*

Why this structure matters

- Mirrors how real product teams work — each section has clear scope and single source of truth
- Gives AI clean, isolated context — when building the hero, it only sees hero specs (no pollution from other sections)
- Makes iteration scalable — refine one section without breaking others

This is specs-driven development applied to AI: clear specs first, build second, then iterate.

Phase 2: AI Creates Documentation

Now that you've done the prep work, it's time to have AI generate the technical documentation. These prompts are designed to be used with Cursor, but work with any AI coding assistant.

Prompt 1: Create Style Guide

The style guide is your single source of truth for all design specs. It's a **living document** that will be updated as you build.

PROMPT:

Based on my moodboard and design preferences in
`website-sections/0.design-moodboard.md`,
create a comprehensive STYLE_GUIDE.md that includes:

- Color palette with CSS variables
- Typography rules (fonts, sizes, weights)
- Component patterns (buttons, cards, etc.)
- Visual effects (glows, gradients, blur)
- Spacing and layout guidelines

This will be our single source of truth for the entire project.

NOTE: This is a LIVING DOCUMENT. As we build sections and learn what works (or doesn't), we will update this guide to reflect our learnings. Add a "Changelog" section at the bottom to track updates.

Prompt 2: Create Project Requirements Document

This document consolidates everything a developer (or AI) needs to know about the project.

PROMPT:

Based on my product brief and all section files in
`website-sections/`, create a PROJECT_REQUIREMENTS.md
that consolidates:

Structure:

1. Project Overview - What, who, goal (from product brief)
2. Tech Stack - Table of technologies and their purpose
3. Dependencies - List of npm packages needed
4. Design System - Just reference STYLE_GUIDE.md, do NOT duplicate it
5. Page Sections - Index table with 1-line descriptions and links to spec files (do NOT duplicate section details, just link to them)
6. File Structure - Recommended folder organization
7. Responsive Requirements - Breakpoints

Rules:

- Do NOT duplicate content from STYLE_GUIDE.md - just reference it
- Do NOT duplicate section details - just create an index with links
- Keep it concise (~80-100 lines)
- This is an overview document, not a copy of everything
- Do NOT include setup steps - those will go in tasks.md

Prompt 3: Create Task List

This creates a step-by-step task list with checkboxes to track progress.

PROMPT:

Based on the STYLE_GUIDE.md and PROJECT_REQUIREMENTS.md, create a tasks.md file with:

- Phase 1: Project setup tasks
- Phase 2: Build each section (reference the section spec files)

Use checkboxes so we can track progress.

Create a Cursor Rule (Important!)

Before building, create this rule so AI automatically updates the style guide when you make changes.

Create file: `.cursor/rules/style-guide-governance.mdc`

```
---
description: Ensures the style guide stays in sync
with implementation
globs:
alwaysApply: true
---

## Style Guide Governance

docs/style-guide.md is the authoritative global
design system.

When implementing a section:
- If a design decision is reusable, document it in
  docs/style-guide.md
- If section-specific, keep it in the section file

If implementation diverges from style guide:
1. If new approach looks/works better, keep it
2. Update style guide to match implementation
3. Add changelog entry noting what changed and why

Never silently diverge from the style guide.
```


Phase 3: Build + Iterate

Now the fun part — building section by section with AI. The key is to iterate: build, review, give feedback, refine.

Prompt 4: Set Up Project (Install Dependencies)

First, tell AI to install all the dependencies from your tasks.md file:

PROMPT:

Start building Phase 1 from tasks.md

AI will install all the dependencies, libraries, fonts, and set up the project structure based on your PROJECT_REQUIREMENTS.md.

 *Wait until all dependencies are installed before moving to building sections.*

Prompt 5: Build First Section

Start with the Navigation + Hero section as it sets the tone for the entire site.

PROMPT:

Build the Hero section from tasks.md

Pro Tips for Building

- Start a NEW chat for each section — keeps context clean and uses fewer tokens
- Commit to GitHub after each section — so you can revert if something breaks
- Take screenshots when giving feedback — paste images directly into Cursor
- Be specific with feedback — "make it 60px" not just "make it bigger"

Real Example: Fixing the 3D Animation

In the tutorial, the 3D angel's colorful marble base wasn't visible. Here's how to fix issues like this:

- Take a screenshot of the current result
- Take a screenshot of the original reference (from your section spec)
- Paste both into Cursor and describe what's different

PROMPT:

Increase the visibility and contrast of the colorful marble base under the angel. Reduce any dark overlays at the bottom of the hero so the base isn't hidden or muted.

[Paste screenshot of current result]
[Paste screenshot of original reference]

💡 *Visual feedback with screenshots is 10x more effective than describing the problem in words.*

Prompt 6: Give Feedback and Iterate

After each section is built, review it and give specific feedback. Here are example prompts:

Visual Adjustments

Examples from the tutorial:

I don't like this harsh separation between the navigation bar and hero section. Make it seamless.

The navigation links shouldn't be in a pill component. Remove that and make the text wider spaced.

When I scroll, I want the navigation bar to have a frosty/glass background effect.

The tag above the headline needs to be inside a component/badge, not just plain text.

Capturing Learnings

I like this spacing better - update the STYLE_GUIDE.md to reflect this as our standard section padding. implementation.

Prompt 7: Build Next Section

Repeat for each remaining section:

PROMPT:

Build the next section from [tasks.md](#) .

Follow the updated STYLE_GUIDE.md and match the patterns established in previous sections.

💡 Repeat prompts 6-8 for each section. The style guide gets BETTER with each section you build!

The Iteration Loop (Key to Success)

This is the core of the framework. For each section:

FOR EACH SECTION

1. AI builds section (following current style guide)
- ↓
2. YOU review and give feedback
- ↓
3. AI refines until you're happy
- ↓
4. AI updates STYLE_GUIDE.md with learnings
- ↓
5. Move to next section (style guide now improved!)

Final File Structure

Here's what your project structure should look like:

```
your-project/
├── .cursor/
│   └── rules/
│       └── style-guide-governance.mdc
├── website-guidelines/
│   ├── product-brief.md           # YOU created
│   ├── STYLE_GUIDE.md            # AI created, UPDATED
│   ├── PROJECT_REQUIREMENTS.md   # AI created
│   └── tasks.md                  # AI created
├── website-sections/
│   ├── inspo-images/             # Your screenshots
│   ├── 0.design-moodboard.md     # Your moodboard
│   ├── 1.navigation-bar.md
│   ├── 2.hero-section.md
│   └── ... (one per section)
├── app/                           # AI generated
│   ├── layout.tsx
│   ├── page.tsx
│   └── globals.css
├── components/                   # AI generated
│   ├── ui/
│   ├── sections/
│   └── three/
├── public/
│   ├── models/
│   ├── textures/
│   └── logos/
```

Quick Start Checklist

Before Your First Prompt to AI

- ☐ Written product brief (what, who, sections)
- ☐ Created moodboard (2-3 inspiration images, colors, fonts)
- ☐ Written content for all sections (headlines, buttons, body)
- ☐ Found and copied component code you want to use

Work with AI

- ☐ Ask AI to create STYLE_GUIDE.md
- ☐ Ask AI to create PROJECT_REQUIREMENTS.md
- ☐ Ask AI to create tasks.md
- ☐ Create .cursor/rules/style-guide-governance.mdc
- ☐ Ask AI to set up the project
- ☐ Build sections one by one (Hero first)

Key Principles to Remember

- Front-load your content work — Write all section content before building
- Moodboard is your design kit — Include found code, not just images
- Style guide is LIVING — Update it after each section with learnings
- Build section by section — Easier to iterate than all at once
- Be specific with feedback — "Make it bigger" → "Make it 60px"
- Capture learnings — Always update style guide after each section
- Commit often — Save your progress to GitHub after each section

Remember: Define → Build → Review → Refine

That loop is everything. Used by real product teams. Works wonders with AI.

The prep work feels like extra effort, but that upfront context — the brief, the moodboards, the style guide, the section specs — that's what separates a \$500 generic website from a \$5,000 premium polished one.

Give AI the same context you'd give a real designer, and premium designs become something you can repeat.