[multiple-ui.md](multiple-ui.md)

```
UI TO BUILD: $ARGUMENTS
OBJECTIVE
Launch 3 concurrent sub-agents in parallel to develop a single UI concept with
different approaches,
Enabling users to compare options and select the most effective solution;
Instructions for each sub-agent


Review the styling specifications in designs/design.json, along with reference mockups
for guidance
Construct a standalone HTML page featuring one interface screen that addresses the
user requirements/brief
Save HTML files to infinte_ui_claude directory using naming convention ui_{n}.html
(Where n must be distinct such as ui_1.html, ui_2.html, etc.)
```

## Extract-design.mdc

```
ANALYZE: the image pasted
OBJECTIVE
Extract a comprehensive and reusable design system from the image pasted, excluding
any specific visual content, to create a JSON reference that developers or AI systems
can use as a styling foundation for consistent UI development.
INSTRUCTIONS


Examine the image pasted to identify:
• Color schemes and palettes
• Font hierarchies and typography standards
• Spacing and margin conventions
• Structural layouts (grid systems, card designs, wrapper elements, etc.)
• Interactive elements (button styles, form inputs, data tables, etc.)
• Visual effects (corner rounding, drop shadows, and additional styling treatments)
Generate a design-system.json file that systematically documents these design
principles and enables consistent visual language reproduction.
Save the JSON output to the designs directory using filename: design.json


REQUIREMENTS


Prioritize extracting scalable design tokens over copying specific interface content
Structure JSON with clear hierarchy and developer-accessible formatting
Include complete color systems, font scaling, and dimensional standards
```

```
Record component variations and interaction states where visible
Follow contemporary design system best practices
Omit any literal text, imagery, or data from source screenshots
Create framework-independent output suitable for any implementation environment
```

## Multiple-ui.mdc

```
WHAT TO BUILD: the text description provided
PURPOSE
Create 3 different versions of the same UI so people can pick which one looks best.
STEPS FOR EACH VERSION

Read the style guide at designs/design.json and look at any example pictures for ideas
Make one HTML page that shows one complete screen based on the text description of
what to build
Save each HTML file in the infinte_ui_cursor folder as ui_1.html, ui_2.html, ui_3.html
(each file must have a different number)
```

## Geo.mdc - You Can Give this to claude and ask it to generate the other ones as well

```
CONFIGURATION: n = 3 files
INSTRUCTIONS
Look at the app design that has been attached.
Create n different versions of that same app, each optimized for a different
geographical region.
Each version should have the same functionality but be tailored for different
locations like:

North America (US/Canada)
Europe (EU regions)
Asia Pacific (Japan, Korea, China, etc.)
Middle East (UAE, Saudi Arabia, etc.)
Latin America (Mexico, Brazil, Argentina, etc.)
Africa (South Africa, Nigeria, etc.)
India (specific cultural preferences)
Nordic Countries (Sweden, Norway, Denmark)
Urban vs Rural (same country, different environments)

Take any additional geographical instructions from the user prompt into account.
FILE MANAGEMENT
```

```
Create a "variations3" folder
Put all the geographical variations inside it


CREATE THE GEOGRAPHICAL DESIGNS
Make n design files:


design1.html (for region 1)
design2.html (for region 2)
design3.html (for region 3)
Continue up to design{n}.html


Each design should be optimized for a different geographical region.
WHAT GOES IN THE FOLDER
/variations/
├── design1.html
├── design2.html
├── design3.html
└── design{n}.html
GEOGRAPHICAL ADAPTATION RULES


Keep the same app functionality
Adapt colors to cultural preferences and meanings
Adjust text direction (left-to-right vs right-to-left)
Modify layouts for cultural reading patterns
Change imagery and icons to be culturally appropriate
Adapt to local legal requirements (GDPR, privacy laws)
Consider local internet infrastructure and speed
Follow regional design trends and preferences


CULTURAL CONSIDERATIONS


Colors: Red means luck in China, danger in West; White means purity in West, mourning
in East
Text Direction: Arabic/Hebrew (right-to-left), Asian (vertical options)
Images: Avoid culturally sensitive imagery, use appropriate representations
Data Formats: Date formats (MM/DD vs DD/MM), currency symbols, number formatting
Privacy: GDPR compliance for EU, different privacy expectations by region


REGIONAL DESIGN PATTERNS


Western: Clean, minimal, lots of white space
```

```
Asian: Information-dense, vibrant colors, detailed interfaces
Middle Eastern: Right-to-left layouts, ornate design elements
Nordic: Clean, functional, accessibility-focused
Latin American: Warm colors, community-focused features


EXAMPLE
If given a to-do list app:

US version: Clean design, productivity-focused, time management features
Japanese version: Detailed categories, group collaboration, respect for hierarchy
Middle Eastern version: Right-to-left text, family/community task sharing, prayer time
integration
```

Infinite-design.mdc

# DESIGN VARIATIONS GENERATOR

**Generate n = 3 files**

## Instructions

Look at the design that has been attached.

Create 3 different versions of that same design. Make each one look different but keep the same basic idea and functionality.

Follow any additional instructions given in the prompt.

## File Management

1. Check if a "variations" folder exists
2. If it exists:
   ○ Rename the reference design to "source.html"
   ○ Delete any other files in the folder
   ○ If "source.html" already exists, delete the old one first
3. If no "variations" folder exists:
   ○ Create the "variations" folder

○ Put the reference design in as "source.html"

## Create Files

Make these files in the variations folder:

- design1.html
- design2.html
- design3.html
- Continue to design{n}.html if more than 3 requested

Each design should be completely different visually but work the same way.

## That's it.

Keep it simple. Make them look different. Follow any extra instructions given.