

Workflow.md

```
# Universal Stitch Code Analysis Workflow
```

```
**For Any Design Generated by Stitch**
```

```
---
```

Workflow Setup

****IMPORTANT:**** Before starting analysis, ensure the following folder structure exists:

1. ****Input Folder:**** `/stitch` - Place all HTML files from Stitch in this folder
2. ****Output Folder:**** `/docs` - All analysis outputs will be saved here as markdown files

```
```bash
```

```
Create required folders
```

```
mkdir -p stitch
```

```
mkdir -p docs
```

```
Place your Stitch HTML files in the stitch folder
```

```
Example: stitch/homepage.html, stitch/products.html, etc.
```

```
```
```

****INPUT SOURCE:**** All HTML files to be analyzed must be placed in the `/stitch` folder.

```
---
```

Phase 1: Design System Extraction

Step 1: Extract Visual Design System

```
```markdown
```

AGENT TASK: Analyze all HTML files from the `/stitch` folder and extract the complete design system.

EXTRACT:

- **Color Palette**: Primary, secondary, background, text, border, and shadow colors
  - **Typography System**: Font families, weights, sizes, line heights
  - **Layout Patterns**: Container widths, padding/margin patterns, border radius
  - **Component Spacing**: Gap values, section padding, consistent spacing units
- ```
```
```

### Step 2: Document and Save Design System

```
```markdown
```

CREATE FILE: `/docs/design-system.md`

DOCUMENT FORMAT:

Design System Analysis

Colors

- Primary: #hex-value
- Secondary: #hex-value
- Background: #hex-value
- Text Primary: #hex-value
- Text Secondary: #hex-value

[... all extracted colors ...]

Typography

- Font Family: [extracted fonts]
- Heading Sizes: h1-h6 with exact rem/px values
- Body Text: [sizes and line heights]

[... all typography details ...]

Layout & Spacing

- Container Max Width: [value]

```
- Section Padding: [pattern]
- Component Margins: [pattern]
- Border Radius: [values used]
[... all layout patterns ...]
...
---
```

Phase 2: Page Structure Mapping

Step 1: Analyze Each Page Structure

```
```markdown
```

AGENT TASK: For each HTML file in the `/stitch` folder, identify and document the complete page structure.

#### ANALYZE:

- **Page Purpose**: Type (homepage, product, etc.), main user goal
  - **Layout Sections**: Header, hero, content areas, sidebars, footer
  - **Interactive Elements**: Buttons, forms, links, hover effects, JS interactions
  - **SEO Metadata**: Page titles, descriptions, meta tags
- ```
...
```

Step 2: Document Page Structures

```
```markdown
```

CREATE FILE: `/docs/page-structures.md`

#### DOCUMENT FORMAT:

##### # Page Structure Analysis

### ## [Page Name 1]

#### ### Purpose

- Type: [page type]
- User Goal: [main goal]

#### ### Sections

1. Header: [description]

2. Hero: [description]  
[... all sections ...]

### ### Interactive Elements

- Buttons: [list all button types]
  - Forms: [list all forms]
- [... all interactive elements ...]

## [Page Name 2]  
[... repeat for all pages ...]  
...

---

## ## Phase 3: Component Library Extraction

### ### Step 1: Identify Reusable Components

```markdown

AGENT TASK: Find patterns that repeat across 2+ HTML files in the `/stitch` folder.

SCAN FOR:

- **Navigation Components**: Headers, menus, breadcrumbs, pagination
- **Content Components**: Cards, buttons, forms, galleries, lists
- **Layout Components**: Grids, containers, modals, accordions
- **Component Variants**: All style variations, states, responsive behaviors

EXTRACTION RULE: If a pattern appears in 2+ files, it's a reusable component.
...

Step 2: Catalog Component Library

```markdown

CREATE FILE: `/docs/component-library.md`

#### DOCUMENT FORMAT:

# Component Library

## ## Buttons

### ### Primary Button

- Classes: `bg-blue-600 text-white hover:bg-blue-700 px-4 py-2 rounded`
- Usage: Main CTAs
- Found in: [list pages]
- States: Default, Hover, Active, Disabled

### ### Secondary Button

[... all button variants ...]

## ## Cards

### ### Product Card

- Structure: image + title + price + button
- Classes: [exact Tailwind/CSS classes]
- Responsive: [breakpoint behaviors]
- Found in: [list pages]

[... all card types ...]

## ## Forms

[... all form components ...]

## ## Navigation

[... all navigation components ...]

...

---

## ## Phase 4: Navigation Flow Documentation

### ### Step 1: Map Site Navigation and User Flows

```markdown

AGENT TASK: Document the complete navigation structure and user journeys.

MAP:

- ****Site Navigation****: Complete sitemap, page hierarchy, menu structure

- ****User Flows****: Button destinations, form submissions, modal triggers
- ****Data Flow****: Data between pages, form requirements, persistent state
- ****API References****: Any endpoints or external services referenced
- ...

Step 2: Create Navigation Documentation

```markdown

CREATE FILE: `/docs/navigation-flow.md`

DOCUMENT FORMAT:

# Navigation Flow

## Site Map

...

Home

```

|— Products
| |— Category A
| |— Category B
|— About
|— Contact

```

...

## User Flows

### Shopping Flow

1. Home → Product List (via nav menu)
  2. Product List → Product Detail (via product card)
  3. Product Detail → Cart (via add to cart button)
- [... all flows ...]

## Interactive Elements Map

| Element           | Location     | Action   | Destination       |
|-------------------|--------------|----------|-------------------|
| "Shop Now" Button | Home Hero    | Navigate | /products         |
| "Add to Cart"     | Product Card | Add Item | Update cart state |

[... all interactions ...]

## ## Data Flow

- Cart State: [how it persists]
- User Session: [what's tracked]
- Form Data: [submission paths]
- ...

---

## ## Phase 5: Implementation Blueprint

### ### Step 1: Generate Implementation Specification

```markdown

AGENT TASK: Create complete implementation guide from all analysis.

SPECIFY:

- **Required Components**: All components with props and priority
- **Design System Constants**: Colors, typography, spacing, animations
- **Forbidden Elements**: What NOT to create or add
- **Build Order**: Core → Page-specific → Enhancements
- ...

Step 2: Create Implementation Blueprint

```markdown

CREATE FILE: `/docs/implementation-blueprint.md`

DOCUMENT FORMAT:

# Implementation Blueprint

## ## Required Components

### ### Priority 1 - Core Components

- [ ] Header with navigation
- [ ] Button component (3 variants)
- [ ] Card component (2 variants)
- [... all components with checkboxes ...]

### ### Priority 2 - Page Components

[... page-specific components ...]

### ### Priority 3 - Enhancements

[... nice-to-have features ...]

## ## Design System Setup

```
``javascript
// Color System
const colors = {
 primary: '#hex-from-analysis',
 // ... all colors
}
```

```
// Typography System
```

```
const typography = {
 // ... from analysis
}
...

```

## ## Forbidden - DO NOT CREATE

- ❌ Custom modal components (none in design)
  - ❌ Additional button styles
  - ❌ New color variations
- [... all restrictions ...]

## ## Build Sequence

1. Setup design system constants
  2. Build layout components
  3. Build navigation
- [... step by step ...]
- ...

---

## ## Phase 6: Validation Checklist



### ### Step 1: Define Validation Criteria

```markdown

AGENT TASK: Create comprehensive validation checklist.

VALIDATE:

- **Design Fidelity**: Colors, typography, spacing, component styles
 - **Functional Requirements**: Navigation, forms, interactions, data flow
 - **Component Compliance**: Required components exist, no extras, props match
 - **Responsive Behavior**: All breakpoints work correctly
- ```

Step 2: Generate Validation Checklist

```markdown

CREATE FILE: `/docs/validation-checklist.md`

DOCUMENT FORMAT:

# Implementation Validation Checklist

## ## Design Validation

### ### Colors

- [ ] Primary color: #exact-hex matches original
  - [ ] Secondary color: #exact-hex matches original
- [... all color checks ...]

### ### Typography

- [ ] Heading sizes match exactly
  - [ ] Font families loaded correctly
- [... all typography checks ...]

## ## Component Validation

### ### Header Component

- [ ] Logo positioned correctly
  - [ ] Navigation items match design
  - [ ] Responsive menu works
- [... component by component ...]

## ## Functional Validation

### ### Navigation

- [ ] Home → Products navigation works
- [ ] All menu links functional
- [... all functional checks ...]

### ## Final Sign-off

- [ ] Pixel-perfect match to original design
- [ ] All interactions work as specified
- [ ] No console errors
- [ ] Responsive on all breakpoints
- ...

---

## ## Workflow Usage Instructions

### ### For Any Stitch Design:

1. **Input:** Place all HTML files from Stitch in the `/stitch` folder
2. **Process:** Run through Phases 1-6 automatically
3. **Output:** Complete implementation specification saved in `/docs` folder
4. **Execute:** Agent follows generated rules exactly

### ### Workflow Benefits:

- **Universal:** Works for any Stitch design (e-commerce, SaaS, portfolio, etc.)
- **Thorough:** Captures every design detail automatically
- **Restrictive:** Prevents agents from adding unauthorized elements
- **Scalable:** Handles single pages or entire applications

### ### Example Command for Agents:

...

"Analyze all HTML files in the /stitch folder using the Universal Workflow.  
Extract the complete design system and component library.  
Save all analysis outputs to the /docs folder."

Generate implementation rules based on your analysis.  
Build the application following only what was found in the code.  
Do not create anything not documented in your analysis."  
'''

This workflow ensures agents understand exactly what to build from any Stitch  
h design, regardless of style, purpose, or complexity.