

# /web-artifacts-builder

**Source:** `~/ .claude/skills/web-artifacts-builder/SKILL.md`

---

**name:** web-artifacts-builder **description:** Suite of tools for creating elaborate, multi-component claude.ai HTML artifacts using modern frontend web technologies (React, Tailwind CSS, shadcn/ui). Use for complex artifacts requiring state management, routing, or shadcn/ui components - not for simple single-file HTML/JSX artifacts. **license:** Complete terms in LICENSE.txt

## Web Artifacts Builder

To build powerful frontend claude.ai artifacts, follow these steps:

1. Initialize the frontend repo using `scripts/init-artifact.sh`
2. Develop your artifact by editing the generated code
3. Bundle all code into a single HTML file using `scripts/bundle-artifact.sh`

4. Display artifact to user
5. (Optional) Test the artifact

**Stack:** React 18 + TypeScript + Vite + Parcel (bundling) + Tailwind CSS + shadcn/ui

# Design & Style Guidelines

VERY IMPORTANT: To avoid what is often referred to as "AI slop", avoid using excessive centered layouts, purple gradients, uniform rounded corners, and Inter font.

## Quick Start

### Step 1: Initialize Project

Run the initialization script to create a new React project:

```
bash scripts/init-artifact.sh <project-name>
cd <project-name>
```

This creates a fully configured project with:

- React + TypeScript (via Vite)
- Tailwind CSS 3.4.1 with shadcn/ui theming system
- Path aliases (@/) configured
- 40+ shadcn/ui components pre-installed
- All Radix UI dependencies included
- Parcel configured for bundling (via .parcelrc)
- Node 18+ compatibility (auto-detects and pins Vite version)

### Step 2: Develop Your Artifact

To build the artifact, edit the generated files. See **Common Development Tasks** below for guidance.

### Step 3: Bundle to Single HTML File

To bundle the React app into a single HTML artifact:

```
bash scripts/bundle-artifact.sh
```

This creates `bundle.html` - a self-contained artifact with all JavaScript, CSS, and dependencies inlined. This file can be directly shared in Claude conversations as an artifact.

**Requirements:** Your project must have an `index.html` in the root directory.

### What the script does:

- Installs bundling dependencies (parcel, @parcel/config-default, parcel-resolver-tspaths, html-inline)
- Creates `.parcelrc` config with path alias support
- Builds with Parcel (no source maps)
- Inlines all assets into single HTML using html-inline

## Step 4: Share Artifact with User

Finally, share the bundled HTML file in conversation with the user so they can view it as an artifact.

## Step 5: Testing/Visualizing the Artifact (Optional)

Note: This is a completely optional step. Only perform if necessary or requested.

To test/visualize the artifact, use available tools (including other Skills or built-in tools like Playwright or Puppeteer). In general, avoid testing the artifact upfront as it adds latency between the request and when the finished artifact can be seen. Test later, after presenting the artifact, if requested or if issues arise.

## Reference

- **shadcn/ui components:** <https://ui.shadcn.com/docs/components>

---

Revision #5

Created 2026-02-18 08:42:05 UTC by John

Updated 2026-06-21 20:01:35 UTC by John