**Absolutely! Here’s a step-by-step audit and checklist to help you build a fully functional wholesaler-style printing site using Gatsby, S&S API, and your local assets.  
I’ll break this into sections:**

* **1️⃣ Core Requirements**
* **2️⃣ API & Data Audit**
* **3️⃣ File & Folder Structure**
* **4️⃣ Gatsby/Netlify/GitHub Setup**
* **5️⃣ Ongoing Checklist**
* **6️⃣ Questions for You**

**1️⃣ Core Requirements for Your Site**

* **<input disabled="" type="checkbox"> Product Pages (with images, descriptions, pricing, color/size options)**
* **<input disabled="" type="checkbox"> Category Pages (e.g., by brand, type, etc.)**
* **<input disabled="" type="checkbox"> Inventory/Stock Info (live or cached)**
* **<input disabled="" type="checkbox"> Order/Cart System (optional for MVP)**
* **<input disabled="" type="checkbox"> Admin/Update Tools (optional for MVP)**
* **<input disabled="" type="checkbox"> API Integration (S&S for products, inventory, etc.)**
* **<input disabled="" type="checkbox"> Hosting & Version Control (Netlify, GitHub)**

**2️⃣ API & Data Audit**

**You Have API Endpoints For:**

* **Products (/v2/styles/)**
* **Categories (/v2/categories/)**
* **Brands (/v2/Brands/)**
* **Single Product by styleID/partNumber/slug**
* **Size Charts (local JSON)**
* **Orders (local JSON, with endpoints for fetching and posting)**
* **Inventory (local JSON, with endpoint for fetching by styleID)**

**Checklist:**

* **<input checked="" disabled="" type="checkbox"> Can fetch all products (but should filter for only brands you want)**
* **<input checked="" disabled="" type="checkbox"> Can fetch categories**
* **<input checked="" disabled="" type="checkbox"> Can fetch brands**
* **<input checked="" disabled="" type="checkbox"> Can fetch single product by ID/slug**
* **<input checked="" disabled="" type="checkbox"> Can fetch size charts**
* **<input checked="" disabled="" type="checkbox"> Can fetch/post orders**
* **<input checked="" disabled="" type="checkbox"> Can fetch inventory**
* **<input disabled="" type="checkbox"> Can fetch only selected brands/products (see below for improvement)**

**3️⃣ File & Folder Structure Audit**

**You Should Have:**

* **<input disabled="" type="checkbox">**[**components**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(React components)**
* **<input disabled="" type="checkbox">**[**pages**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(Gatsby pages)**
* **<input disabled="" type="checkbox">**[**templates**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(Gatsby page templates, e.g., ProductPageTemplate.jsx)**
* **<input disabled="" type="checkbox">**[**data**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(local JSON: products.json, all\_styles\_raw.json, sizeCharts.json, orders.json, inventory.json)**
* **<input disabled="" type="checkbox"> static/product-images/ (organized by brand/product, with product-image-map.json in each)**
* **<input disabled="" type="checkbox">**[**.gitignore**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(ignoring**[**node\_modules**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**, .cache, public, large data/image folders not needed in git)**
* **<input disabled="" type="checkbox">**[**gatsby-config.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(with correct plugins and sources)**
* **<input disabled="" type="checkbox">**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(with correct logic for sourcing nodes and creating pages)**
* **<input disabled="" type="checkbox">**[**gatsby-ssr.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(for context providers/global styles)**
* **<input disabled="" type="checkbox"> [netlify.toml](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") (optional, for Netlify config)**

**4️⃣ Gatsby/Netlify/GitHub Setup Audit**

* **<input checked="" disabled="" type="checkbox"> GitHub repo (for version control)**
* **<input checked="" disabled="" type="checkbox"> Netlify account (for hosting)**
* **<input disabled="" type="checkbox"> Netlify site connected to GitHub repo**
* **<input disabled="" type="checkbox"> Environment variables set in Netlify (for S&S API keys)**
* **<input disabled="" type="checkbox"> Build command set to gatsby build**
* **<input disabled="" type="checkbox"> Publish directory set to public**

**5️⃣ Ongoing Checklist**

**A. Data & API**

* **<input disabled="" type="checkbox"> Are all API endpoints working (test with Postman or browser)?**
* **<input disabled="" type="checkbox"> Are you filtering products by allowed brands in**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**?**
* **<input disabled="" type="checkbox"> Are local JSON files up to date and in the right format?**
* **<input disabled="" type="checkbox"> Are image folders organized and mapped in product-image-map.json?**

**B. Gatsby**

* **<input disabled="" type="checkbox"> Does gatsby develop work with your current data?**
* **<input disabled="" type="checkbox"> Are product/category pages being created as expected?**
* **<input disabled="" type="checkbox"> Are images showing up on product pages?**
* **<input disabled="" type="checkbox"> Are size charts and inventory showing up?**
* **<input disabled="" type="checkbox"> Is the site fast and responsive with your current dataset?**

**C. Deployment**

* **<input disabled="" type="checkbox"> Can you build and deploy to Netlify without errors?**
* **<input disabled="" type="checkbox"> Are environment variables set in Netlify for API access?**
* **<input disabled="" type="checkbox"> Is the live site working as expected?**

**D. Version Control**

* **<input disabled="" type="checkbox"> Is your**[**.gitignore**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**keeping large/unneeded files out of git?**
* **<input disabled="" type="checkbox"> Are you pushing only what’s needed to GitHub?**

**6️⃣ Questions for You**

1. **Which brands/products do you want to start with?  
   (You said Gildan, Next Level, M&O, Bella+Canvas, Independent Trading Co.—confirm or add more.)**
2. **Do you want to use live S&S API data, or start with a local JSON export?  
   (Live API is best for inventory, but local is faster for development.)**
3. **Do you have all images for your selected products in static/product-images/?  
   (If not, do you want to use S&S image URLs, or upload your own?)**
4. **Do you want to support live inventory/stock info, or is static data OK for now?**
5. **Do you want to support ordering/cart at launch, or just product browsing?**
6. **Is your Netlify site already connected to your GitHub repo?**

**Next Steps**

1. **Answer the questions above.**
2. **Let me know if you want a script or code to filter products by brand in your API fetch.**
3. **Tell me if you want a sample folder structure or**[**.gitignore**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**template.**
4. **If you want, I can walk you through testing each API endpoint and checking your Gatsby build step by step.**

**Let’s do this together!  
Reply with your answers and any files/folders you want me to check, and I’ll guide you through the next steps and update your checklist as we go**

**i believe we are missing inventory.json., product-image-map.json. we are having issues running gatsby develop and having a working homepage pop up locally. i was working last week and it's stopped due to some changes we've made. Images are not showing up on product pages, although we have tested some product images retreived via api and it was working, just not on a proper page. we messed around for a week to no avail and then started trying to get the images to populate on screen with a static folder of images. But when we did that, we couldn't upload to github due to size constraints. Now we are going to trim the fat so to speak and just use the products we want and use api live data fetching to produce the pages. i'm hoping a streamlined product flow will work. I'll post the contents of each "folder" like data, and src, etc. you review and assess code and check off the list the parts we reviewed and cleared for launch. we will then verify steps 4 through 6.  
  
Perfect! This is a great approach. Here’s how we’ll proceed:**

**Your Streamlined Audit & Launch Checklist**

**Step 1: Folder & Data Audit**

* **<input disabled="" type="checkbox"> data/ (should contain: products.json, inventory.json, sizeCharts.json, etc.)**
* **<input disabled="" type="checkbox"> static/product-images/ (should contain only images for selected products, with product-image-map.json in each product folder)**
* **<input disabled="" type="checkbox"> src/ (components, templates, pages)**
* **<input disabled="" type="checkbox">**[**gatsby-config.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(correct plugins and sources)**
* **<input disabled="" type="checkbox">**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(fetches only selected brands/products, creates pages)**
* **<input disabled="" type="checkbox"> .gitignore (already reviewed and ✅ good)**

**Step 2: Functionality Audit**

* **<input disabled="" type="checkbox"> gatsby develop runs without errors**
* **<input disabled="" type="checkbox"> Homepage loads locally**
* **<input disabled="" type="checkbox"> Product pages generate and display images**
* **<input disabled="" type="checkbox"> API fetch for products, inventory, and images works**
* **<input disabled="" type="checkbox"> No large, unnecessary files in repo**

**Step 3: Deployment Audit**

* **<input disabled="" type="checkbox"> Netlify connected to GitHub**
* **<input disabled="" type="checkbox"> Environment variables set in Netlify**
* **<input disabled="" type="checkbox"> Site builds and deploys successfully**

**Step 4: Final Review**

* **<input disabled="" type="checkbox"> Live site works as expected**
* **<input disabled="" type="checkbox"> All required features present**
* **<input disabled="" type="checkbox"> Performance is acceptable**

**How We’ll Work**

1. **You post the contents of each folder (data, static, src, etc.) one at a time.**
2. **I’ll review and check off each part, giving you feedback and fixes as needed.**
3. **Once all folders are reviewed and fixed, we’ll move to functionality and deployment checks.**
4. **We’ll finish with a final review and launch checklist.**

**What To Post Next**

* **Start with your data folder (list all files and a sample of their contents, especially products.json and inventory.json).**
* **Then move to static/product-images (list folders/files and a sample product-image-map.json).**
* **Then src (list components, templates, and pages).**
* **Then**[**gatsby-config.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**and**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**.**

**Let’s do this step by step. Post your**[**data**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**folder contents first, and I’ll review and check it off**

**Data Folder Audit**

**What you have:**

* **✅ [products.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") (contains product entries, but all are "Unknown" and have empty images/prices)**
* **✅ [all\_styles\_raw.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") (contains many objects, but only a sustainableStyle property)**
* **✅**[**sizeCharts.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**orders.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**newsletter.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**faqs.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**discounts.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**customers.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**contactMessages.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**carts.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**blog.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**audit.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**analytics.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**reviews.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**,**[**userRoles.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(all empty arrays)**
* **✅ [driveFiles.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") (Google Drive file metadata, not used by Gatsby directly)**

**What’s missing or needs attention:**

* **⚠️ inventory.json is missing (needed if you want to show live/cached inventory)**
* **⚠️ No product-image-map.json files (needed for mapping images to products if using local images)**
* **⚠️ [products.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") is placeholder data (all products are "Unknown", no real brand, price, or images)**
* **⚠️ [all\_styles\_raw.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") is not in a usable format for product display (missing product details, just a flag)**

**Checklist for Data Folder**

| **File** | **Status** | **Notes/Action Needed** |
| --- | --- | --- |
| [**products.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **⚠️ Placeholder** | **Needs real product data (brand, name, price, images)** |
| [**all\_styles\_raw.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **⚠️ Incomplete** | **Needs full product info if used for product pages** |
| **inventory.json** | **❌ Missing** | **Needed for inventory display** |
| [**sizeCharts.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **⚠️ Empty** | **Add size chart data if you want to show charts** |
| **product-image-map.json** | **❌ Missing** | **Needed if using local images** |
| **...other files** | **⚠️ Empty** | **OK if not needed for MVP** |

**Next Steps**

1. **Decide your data source:**
   * **If using the S&S API, you don’t need to maintain [products.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") or [all\_styles\_raw.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") for live products, but you do need to filter for your chosen brands in your API fetch.**
   * **If using local data, you must fill [products.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") with real product info and add product-image-map.json for images.**
2. **For API-driven site:**
   * **Update your**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**to fetch only your chosen brands (Gildan, Next Level, M&O, Bella+Canvas, Independent Trading Co.).**
   * **Make sure your product page template expects the API data structure.**
3. **For local images:**
   * **For each product you want to show, create a folder in static/product-images/brand/style/ and add a product-image-map.json mapping color names to image files.**
4. **For inventory:**
   * **If you want to show inventory, create inventory.json or fetch inventory live from the API.**

**Questions for You**

1. **Do you want to use only live API data for products, or do you want to supplement with local JSON?**
2. **Do you want to use S&S image URLs, or do you want to use your own local images?**
3. **Do you want to show inventory/stock info on product pages?**

**Once you answer, I’ll give you the next checklist and code/folder recommendations.  
Ready for the next folder (static/product-images) when you are**

**Searched static for "i want to use live api data, and dont want to use local data if i can help it but i know there will be times when we do use static images. i have the whole s&s inventory loaded in the static folder. i want to use the products from a few select brands, and have live inventory/stock/pricing updates. let's focus on that for now. is using s&s urls the standard practice? if so let's do it. sharing the static folder, and then i'll share API", 58 results**

**Excellent, your plan is clear and aligns with best practices for a scalable, maintainable site:**

**Your Approach (Best Practice for Wholesaler Sites)**

* **Live API data for products, inventory, and pricing (always up-to-date, less local data to manage)**
* **Use S&S image URLs (standard for most integrations; saves bandwidth and storage, always current)**
* **Local static images only as fallback (for custom, missing, or special products)**

**Review of Your**[**static**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**Folder**

* **You have a well-organized static image structure with**[**product-image-map.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**for each product. This is great for fallback or custom images.**
* **For a live API-driven site, you do not need to load all S&S images locally—just use the URLs provided by the API for each product/color.**
* **Keep your static images for:**
  + **Custom products not in S&S**
  + **Fallbacks if an S&S image is missing or broken**
  + **Special marketing/hero images**

**Is Using S&S Image URLs Standard?**

**Yes!  
Most wholesalers and print-on-demand sites use the image URLs provided by the supplier’s API.**

* **Pros:**
  + **Always up-to-date images**
  + **No need to store or update thousands of files**
  + **Less bandwidth/storage on your server**
* **Cons:**
  + **If S&S changes their URLs or removes an image, you lose that image (rare, but possible)**
  + **Slightly slower load if S&S servers are slow (but they’re usually fast)**

**How to Proceed**

**1. Update Your Gatsby Data Layer**

* **In**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**, fetch only the brands you want from the S&S API (Gildan, Next Level, M&O, Bella+Canvas, Independent Trading Co.).**
* **For each product, store the S&S image URLs (these are usually in the API response).**

**2. Product Page Template**

* **Display images using the S&S URLs from the API.**
* **If a product/color is missing an image, fallback to your local static image (using your**[**product-image-map.json**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**).**

**3. Inventory & Pricing**

* **Use the S&S API endpoints for live inventory and pricing on product pages.**

**4. Static Images**

* **Only reference your local static images if you want to override or supplement S&S images.**

**Checklist for This Approach**

| **Step** | **Status/Action Needed** |
| --- | --- |
| **Use S&S API for products** | **Update**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**to fetch only select brands** |
| **Use S&S image URLs** | **Use API image URLs in product templates** |
| **Fallback to static images** | **Use local images only if S&S image is missing** |
| **Live inventory/pricing** | **Use S&S API endpoints for inventory/pricing** |
| **Clean up static folder** | **Keep only custom/fallback images** |

**Next Steps**

1. **Share your API fetch logic or the S&S API response for a product (so I can show you exactly how to use the image URLs).**
2. **Update your Gatsby product template to use the S&S image URLs, with a fallback to local images.**
3. **Update your**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**to only create nodes/pages for your selected brands.**

**Let’s move to your API code next!  
Share your API fetch logic or a sample S&S API product response, and I’ll guide you through the exact code and template changes needed.**

**Great summary! Here’s a clear checklist and guidance based on your goals and what you’ve shared so far:**

**✅ Your Streamlined Launch Checklist**

**1. Data & API Setup**

* **<input disabled="" type="checkbox"> Create [inventory.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**
  + **For now, you can use a placeholder or fetch live from S&S API.**
* **<input disabled="" type="checkbox"> Add placeholder/sample data to JSON files**
  + **Especially for [products.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [sizeCharts.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), etc., if you want to test local features.**
* **<input disabled="" type="checkbox"> Update**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**to use S&S API**
  + **Fetch only your selected brands (Gildan, Next Level, M&O, Bella+Canvas, Independent Trading Co.).**
  + **Use S&S API for live product, inventory, and pricing data.**

**2. Static Folder & Images**

* **<input disabled="" type="checkbox"> Clean up static/product-images/**
  + **Keep only custom/fallback images.**
  + **Remove bulk S&S images if you’re using S&S URLs.**
* **<input disabled="" type="checkbox"> Use S&S image URLs in product templates**
  + **Display images directly from the S&S API response.**
  + **Fallback to local images only if needed.**

**3. Product Page & Inventory**

* **<input disabled="" type="checkbox"> Update product page templates**
  + **Use S&S API data for product info, images, inventory, and pricing.**
* **<input disabled="" type="checkbox"> Use S&S API endpoints for live inventory/pricing**
  + **Fetch and display real-time stock and price info.**

**4. General Site Health**

* **<input disabled="" type="checkbox"> Ensure all API endpoints are working**
  + **Test /api/list-products, /api/get-products, /api/list-brands, /api/get-inventory, etc.**
* **<input disabled="" type="checkbox"> Homepage and product/category pages load locally (gatsby develop)**
* **<input disabled="" type="checkbox"> No large, unnecessary files in repo (**[**.gitignore**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**is up to date)**

**📝 Next Steps for You**

1. **Create [inventory.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") (even as an empty array or with a sample entry for now).**
2. **Add at least one sample entry to each empty JSON file in**[**data**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**(for testing).**
3. **Update**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**:**
   * **Fetch only your selected brands from S&S API.**
   * **Use S&S image URLs in the nodes you create.**
4. **Clean up your static/product-images/ folder (keep only what you need for fallback/custom).**
5. **Update your product page template to use S&S image URLs, with a fallback to local images if needed.**
6. **Test your API endpoints (use Postman or your browser).**

**❓ Questions for You**

* **Do you want a sample [inventory.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [products.json](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") structure for placeholders?**
* **Would you like a code sample for filtering S&S API results by brand in**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**?**
* **Do you want a sample React component for displaying S&S images with a local fallback?**
* **Are there any custom products or images you want to always use locally?**

**Once you complete the above, share your updated**[**gatsby-node.js**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**and product template code for review.  
Let me know which code samples or templates you want next**

**Audit: utils, templates, hooks, helpers, context**

**What’s Good**

* **Templates: You have templates for product, category, categories, and brand pages.**
* **Hooks: [useSpecs](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") fetches live S&S specs; [useModalToggle](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") is a clean modal state hook.**
* **Helpers: Category slug map and print quote calculator are present.**
* **Context: Cart and notification providers are implemented and ready for use.**
* **Utils: Square SDK loader is present for payment integration.**

**What Needs Attention**

* **ProductPageTemplate and other templates/components:**
  + **Image URLs: Still use local paths like /images/Style/${style}\_fm.jpg.  
    Action: Update to use S&S API image URLs, with fallback to local/static if needed.**
  + **Data Source: ProductPageTemplate expects [productsJson](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") from GraphQL.  
    Action: If using live API, update to use API data (via context, props, or page query).**
* **CategoryPageTemplate:**
  + **Fetches from /api/get-products.  
    Action: Ensure this endpoint returns filtered S&S API data for your selected brands.**
* **BrandPageTemplate:**
  + **Uses [allProductsJson](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") GraphQL query.  
    Action: Update to use API data or ensure your API endpoint supports brand filtering.**
* **CartContext:**
  + **Uses [styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") for cart items.  
    Action: Ensure all product objects have [styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") from the S&S API.**
* **Fallbacks:**
  + **Add a /placeholder.png image for missing product images.**

**📝 TO-DO List (Updated)**

* **<input disabled="" type="checkbox"> Update all image references in templates/components to use S&S API image URLs, with fallback to local/static images.**
* **<input disabled="" type="checkbox"> Update data fetching in templates to use live S&S API data, not just local GraphQL queries.**
* **<input disabled="" type="checkbox"> Ensure /api/get-products and similar endpoints return filtered, up-to-date S&S API data.**
* **<input disabled="" type="checkbox"> Add /placeholder.png for missing images.**
* **<input disabled="" type="checkbox"> Ensure all cart/product objects use S&S API fields ([styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), etc.).**
* **<input disabled="" type="checkbox"> Test category/brand/product pages for correct data and image display.**
* **<input disabled="" type="checkbox"> Test Square payment integration (when ready).**

**Next Steps**

1. **Update image logic in all templates/components.**
2. **Update data fetching in templates to use S&S API data.**
3. **Verify**[**api**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**endpoints return correct, filtered S&S data.**
4. **Share your**[**pages**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**folder and any other key files for review.**
5. **Let me know if you want code samples for any of the above!**

**You’re making excellent progress!  
Ready for the next folder or any questions you hav**

**Audit: utils, templates, hooks, helpers, context**

**What’s Good**

* **Templates: You have templates for product, category, categories, and brand pages.**
* **Hooks: [useSpecs](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") fetches live S&S specs; [useModalToggle](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") is a clean modal state hook.**
* **Helpers: Category slug map and print quote calculator are present.**
* **Context: Cart and notification providers are implemented and ready for use.**
* **Utils: Square SDK loader is present for payment integration.**

**What Needs Attention**

* **ProductPageTemplate and other templates/components:**
  + **Image URLs: Still use local paths like /images/Style/${style}\_fm.jpg.  
    Action: Update to use S&S API image URLs, with fallback to local/static if needed.**
  + **Data Source: ProductPageTemplate expects [productsJson](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") from GraphQL.  
    Action: If using live API, update to use API data (via context, props, or page query).**
* **CategoryPageTemplate:**
  + **Fetches from /api/get-products.  
    Action: Ensure this endpoint returns filtered S&S API data for your selected brands.**
* **BrandPageTemplate:**
  + **Uses [allProductsJson](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") GraphQL query.  
    Action: Update to use API data or ensure your API endpoint supports brand filtering.**
* **CartContext:**
  + **Uses [styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") for cart items.  
    Action: Ensure all product objects have [styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") from the S&S API.**
* **Fallbacks:**
  + **Add a /placeholder.png image for missing product images.**

**📝 TO-DO List (Updated)**

* **<input disabled="" type="checkbox"> Update all image references in templates/components to use S&S API image URLs, with fallback to local/static images.**
* **<input disabled="" type="checkbox"> Update data fetching in templates to use live S&S API data, not just local GraphQL queries.**
* **<input disabled="" type="checkbox"> Ensure /api/get-products and similar endpoints return filtered, up-to-date S&S API data.**
* **<input disabled="" type="checkbox"> Add /placeholder.png for missing images.**
* **<input disabled="" type="checkbox"> Ensure all cart/product objects use S&S API fields ([styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), etc.).**
* **<input disabled="" type="checkbox"> Test category/brand/product pages for correct data and image display.**
* **<input disabled="" type="checkbox"> Test Square payment integration (when ready).**

**Next Steps**

1. **Update image logic in all templates/components.**
2. **Update data fetching in templates to use S&S API data.**
3. **Verify [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints return correct, filtered S&S data.**
4. **Share your**[**pages**](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**folder and any other key files for review.**
5. **Let me know if you want code samples for any of the above!**

**You’re making excellent progress!  
Ready for the next folder or any questions you hav**

**Pages & Styles Audit**

**What’s Good**

* **Pages are well-structured**: You have clear separation for products, brands, categories, cart, checkout, admin, and static content (about, faq, blog, etc.).
* **API-driven**: Most product/category/brand pages fetch data from [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints, which is ideal for live S&S integration.
* **Admin tools**: You have an admin print-orders page with CSV export and password protection.
* **Cart & checkout**: Cart and checkout flows are present and styled.
* **Styles**: CSS is modular and matches your brand’s look.

**What Needs Attention**

* **Image URLs**:
  + Product and saved items pages sometimes use local image paths or expect a field like [product.image](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
  + **Action:** Update all product image references to use S&S API image URLs, with fallback to local/static images or a placeholder.
* **API endpoints**:
  + Ensure /api/list-products, /api/get-products, /api/list-brands, /api/list-categories, /api/get-product, etc. are implemented and return the correct, filtered S&S data.
* **Fallbacks**:
  + Add /placeholder.png for missing images.
* **Static images**:
  + Clean up static/product-images/ to keep only custom/fallback images.
* **404 and under-construction pages**:
  + These are fine for now, but plan to fill them in before launch.
* **Security**:
  + Never commit real admin passwords to your repo. Use environment variables for sensitive data.

**📝 Updated TO-DO List**

**A. Data & API**

* <input disabled="" type="checkbox"> Create inventory.json (placeholder or fetch live from S&S API if needed for fallback).
* <input disabled="" type="checkbox"> Add at least one sample entry to each empty JSON file in [data](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) for testing.
* <input disabled="" type="checkbox"> Update [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints to fetch and filter S&S API data for your selected brands.
* <input disabled="" type="checkbox"> Ensure /api/get-product returns all needed fields (including S&S image URLs).

**B. Gatsby & Components**

* <input disabled="" type="checkbox"> Update all product image references in components and templates to use S&S API image URLs, with fallback to local/static images or /placeholder.png.
* <input disabled="" type="checkbox"> Update data fetching in templates to use live S&S API data, not just local GraphQL queries.
* <input disabled="" type="checkbox"> Add /placeholder.png for missing images.
* <input disabled="" type="checkbox"> Ensure all cart/product objects use S&S API fields ([styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), etc.).
* <input disabled="" type="checkbox"> Test category/brand/product pages for correct data and image display.
* <input disabled="" type="checkbox"> Test Square payment integration (when ready).

**C. Static Folder**

* <input disabled="" type="checkbox"> Clean up static/product-images/ to keep only custom/fallback images.

**D. General Site Health**

* <input disabled="" type="checkbox"> Ensure all API endpoints are working (test with Postman or browser).
* <input disabled="" type="checkbox"> Homepage and product/category pages load locally (gatsby develop).
* <input disabled="" type="checkbox"> No large, unnecessary files in repo ([.gitignore](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is up to date).

**E. Security**

* <input disabled="" type="checkbox"> Move any hardcoded passwords or secrets to environment variables.

**How the Website Will Operate (Memory Bank Update)**

* **Product, category, and brand pages**: Fetch live data from S&S API via your [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints.
* **Images**: Use S&S image URLs from the API. Fallback to local images or /placeholder.png if needed.
* **Cart/checkout**: Use S&S product data for pricing and inventory, Square for payments.
* **Admin**: Print order management and CSV export via protected admin page.
* **Static content**: About, FAQ, blog, etc. are static React pages.
* **Fallbacks**: Local images and JSON only used if S&S API is unavailable or for custom products.

**Next Steps**

1. **Update**[api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**endpoints** to fetch and filter S&S API data for your selected brands.
2. **Update all product image logic** in components/templates to use S&S image URLs.
3. **Add /placeholder.png** to your static assets.
4. **Clean up static images** to keep only what’s needed.
5. **Test your site locally** (gatsby develop) and fix any broken images or missing data.
6. **Continue sharing any other folders/files you want reviewed.**

**Let me know when you’re ready for the next review, or if you want code samples for any of the above steps**

**Pages & Styles Audit**

**What’s Good**

* **Pages are well-structured**: You have clear separation for products, brands, categories, cart, checkout, admin, and static content (about, faq, blog, etc.).
* **API-driven**: Most product/category/brand pages fetch data from [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints, which is ideal for live S&S integration.
* **Admin tools**: You have an admin print-orders page with CSV export and password protection.
* **Cart & checkout**: Cart and checkout flows are present and styled.
* **Styles**: CSS is modular and matches your brand’s look.

**What Needs Attention**

* **Image URLs**:
  + Product and saved items pages sometimes use local image paths or expect a field like [product.image](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
  + **Action:** Update all product image references to use S&S API image URLs, with fallback to local/static images or a placeholder.
* **API endpoints**:
  + Ensure /api/list-products, /api/get-products, /api/list-brands, /api/list-categories, /api/get-product, etc. are implemented and return the correct, filtered S&S data.
* **Fallbacks**:
  + Add /placeholder.png for missing images.
* **Static images**:
  + Clean up static/product-images/ to keep only custom/fallback images.
* **404 and under-construction pages**:
  + These are fine for now, but plan to fill them in before launch.
* **Security**:
  + Never commit real admin passwords to your repo. Use environment variables for sensitive data.

**📝 Updated TO-DO List**

**A. Data & API**

* <input disabled="" type="checkbox"> Create inventory.json (placeholder or fetch live from S&S API if needed for fallback).
* <input disabled="" type="checkbox"> Add at least one sample entry to each empty JSON file in [data](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) for testing.
* <input disabled="" type="checkbox"> Update [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints to fetch and filter S&S API data for your selected brands.
* <input disabled="" type="checkbox"> Ensure /api/get-product returns all needed fields (including S&S image URLs).

**B. Gatsby & Components**

* <input disabled="" type="checkbox"> Update all product image references in components and templates to use S&S API image URLs, with fallback to local/static images or /placeholder.png.
* <input disabled="" type="checkbox"> Update data fetching in templates to use live S&S API data, not just local GraphQL queries.
* <input disabled="" type="checkbox"> Add /placeholder.png for missing images.
* <input disabled="" type="checkbox"> Ensure all cart/product objects use S&S API fields ([styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), etc.).
* <input disabled="" type="checkbox"> Test category/brand/product pages for correct data and image display.
* <input disabled="" type="checkbox"> Test Square payment integration (when ready).

**C. Static Folder**

* <input disabled="" type="checkbox"> Clean up static/product-images/ to keep only custom/fallback images.

**D. General Site Health**

* <input disabled="" type="checkbox"> Ensure all API endpoints are working (test with Postman or browser).
* <input disabled="" type="checkbox"> Homepage and product/category pages load locally (gatsby develop).
* <input disabled="" type="checkbox"> No large, unnecessary files in repo ([.gitignore](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is up to date).

**E. Security**

* <input disabled="" type="checkbox"> Move any hardcoded passwords or secrets to environment variables.

**How the Website Will Operate (Memory Bank Update)**

* **Product, category, and brand pages**: Fetch live data from S&S API via your [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints.
* **Images**: Use S&S image URLs from the API. Fallback to local images or /placeholder.png if needed.
* **Cart/checkout**: Use S&S product data for pricing and inventory, Square for payments.
* **Admin**: Print order management and CSV export via protected admin page.
* **Static content**: About, FAQ, blog, etc. are static React pages.
* **Fallbacks**: Local images and JSON only used if S&S API is unavailable or for custom products.

**Next Steps**

1. **Update**[api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**endpoints** to fetch and filter S&S API data for your selected brands.
2. **Update all product image logic** in components/templates to use S&S image URLs.
3. **Add /placeholder.png** to your static assets.
4. **Clean up static images** to keep only what’s needed.
5. **Test your site locally** (gatsby develop) and fix any broken images or missing data.
6. **Continue sharing any other folders/files you want reviewed.**

**Let me know when you’re ready for the next review, or if you want code samples for any of the above steps**

Perfect! Here’s an **updated audit and running TO-DO list** based on your latest batch of files. You’re very close to a launch-ready, maintainable, and scalable site.

**✅ Audit Summary (New Files)**

**Build & Deployment**

* **Netlify & GitHub Actions:**
  + You have CI/CD set up for Netlify deploys and lint/tests on push.
  + [.env.development](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and secrets are present (be sure not to commit real secrets to public repos).
* **Netlify Functions:**
  + [process-payment.js](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is set up for Square payments (sandbox mode for now).
* **Gatsby Config:**
  + [gatsby-config.js](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [gatsby-node.js](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) are set up for S&S API integration and local data fallback.
* **Scripts:**
  + [fetchAllStyles.js](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [updateSustainableStyle.js](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) for data management.
* **.gitignore & .gitattributes:**
  + Large files and sensitive files are ignored; LFS is set up for images/fonts.

**📝 Running TO-DO List (Updated)**

**A. Data & API**

* <input disabled="" type="checkbox"> **inventory.json:** Create or fetch live from S&S API if needed for fallback.
* <input disabled="" type="checkbox"> **Sample Data:** Add at least one sample entry to each empty JSON file in [data](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) for testing.
* <input disabled="" type="checkbox"> **API Endpoints:**
  + Update [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints to fetch and filter S&S API data for your selected brands.
  + Ensure /api/get-product returns all needed fields (including S&S image URLs).

**B. Gatsby & Components**

* <input disabled="" type="checkbox"> **Image URLs:**
  + Update all product image references in components and templates to use S&S API image URLs, with fallback to local/static images or /placeholder.png.
* <input disabled="" type="checkbox"> **Data Fetching:**
  + Update data fetching in templates to use live S&S API data, not just local GraphQL queries.
* <input disabled="" type="checkbox"> **Placeholder Image:**
  + Add /placeholder.png for missing images.
* <input disabled="" type="checkbox"> **Cart/Product Objects:**
  + Ensure all cart/product objects use S&S API fields ([styleID](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), etc.).
* <input disabled="" type="checkbox"> **Test Pages:**
  + Test category/brand/product pages for correct data and image display.
* <input disabled="" type="checkbox"> **Test Square Payment:**
  + Test Square payment integration (when ready).

**C. Static Folder**

* <input disabled="" type="checkbox"> **Clean Up Images:**
  + Clean up static/product-images/ to keep only custom/fallback images.

**D. General Site Health**

* <input disabled="" type="checkbox"> **API Endpoint Testing:**
  + Ensure all API endpoints are working (test with Postman or browser).
* <input disabled="" type="checkbox"> **Local Development:**
  + Homepage and product/category pages load locally (gatsby develop).
* <input disabled="" type="checkbox"> **Repo Size:**
  + No large, unnecessary files in repo ([.gitignore](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is up to date).

**E. Security**

* <input disabled="" type="checkbox"> **Secrets:**
  + Move any hardcoded passwords or secrets to environment variables.
  + Never commit real secrets to public repos.

**How the Website Will Operate (Memory Bank Update)**

* **Product, category, and brand pages:** Fetch live data from S&S API via your [api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") endpoints.
* **Images:** Use S&S image URLs from the API. Fallback to local images or /placeholder.png if needed.
* **Cart/checkout:** Use S&S product data for pricing and inventory, Square for payments.
* **Admin:** Print order management and CSV export via protected admin page.
* **Static content:** About, FAQ, blog, etc. are static React pages.
* **Fallbacks:** Local images and JSON only used if S&S API is unavailable or for custom products.

**Next Steps**

1. **Show image and fonts folders** (if you want a final audit).
2. **Update**[api](vscode-file://vscode-app/c:/Users/Stacey/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**endpoints and image logic** as described above.
3. **Test your site locally and on Netlify.**
4. **Let me know if you want code samples for any of the above steps!**

**You’re almost there!  
Ready for the next batch or any questions you have**