

Shopify+ FE Skills Test

What We Expect

In order for us to validate your code, please create a GitHub Repo with your Work. Follow Shopify's guide for [setting up a development store](#) and use the pre-installed Debut theme as the basis for your work. You'll want to set-up [Themekit](#) so you can develop locally and deploy to your store.

Populate the store with some dummy data (we recommend the [Shopify Developer Tools](#), importing some data through CSV, or you can do it manually). You'll need at least **4 products and 1 collection**.

Commit the theme to the repo as an initial commit, then **commit your work at least once for each step** and push it up to GitHub for our review. Include a [README.md](#) in the repo with a link to your Shopify Store and the storefront password.

The skills test consists of creating a new Shopify Section. It will cover the basics of Liquid/HTML, CSS/SCSS and Javascript required for Shopify Front End development.

We like to keep our SCSS/CSS and Javascript separate from our Liquid, but since we're working within a pre-existing theme, it is okay to write all your Liquid, CSS, JavaScript directly within the `.liquid` section file, wrapped in appropriate tags.

Skills Test

Part 1: Create a new Section

We need a new Featured Products section in the theme to show off some products.

Requirements

- Create a brand new dynamic section for the homepage called **Feature Products**
- The section should be able to be moved up and down the page using the theme settings
- The user should be able to specify a **collection** in the theme settings and the first 4 products of the collection should display in a row.
- The user should be able to specify a **title** for the section which appears above the product row.
- For each product, display the following:
 - Main Product Image
 - Product Title
 - Product Price
- Clicking on the Product Image or Product Title should link the user to the Product Page.
- On desktop, display the items 4 across. On tablet and below, display them 2 x 2.
- The inherited styles from the theme are fine for the text elements.

Technical Approach

- Refer to Shopify's documentation and instructions for setting up a section with a settings schema.
- In particular, we are looking for a dynamic section for the Homepage, not a static section.
- Resources
 - <https://www.shopify.ca/partners/blog/how-to-create-your-first-shopify-theme-section>
 - <https://shopify.dev/docs/themes/settings>

Part 2: Add to Cart Button Styling

We want to encourage shoppers to add the featured products to cart right away. We'll achieve that by adding a new button to our sections and giving them some nice styling.

Requirements

- Button styling should match the first pairing from the mock ups provided below as closely as possible
- Button styling should change when the user hovers over them
- Button text should say: “Add to Cart”
- No animation/transitions required

Mock Ups

BUTTONS



Technical Approach

- We prefer the use of BEM class naming conventions to handle the custom styling
- We prefer the use of SCSS over CSS

Part 3: Add to Cart Button Javascript

We want these new buttons to do something. Let's use the Shopify Cart API to add the product to cart when the user clicks on the button.

Requirements

- Clicking the “Add to Cart” button should add 1 of the product's **selected or first available variants** to cart
- The page should not reload
- The header cart does **not** need to update at this time
- The updated Cart response received back from the API should log out to the browser console

Technical Approach

- Make a REST API call to POST to the Shopify `/cart/add.js` endpoint available on the front end
- Use Liquid to get the product's variant ID which needs to be passed in the API call
- Set the quantity to add to 1 directly in your Javascript
- JavaScript can be added directly in your `.liquid` section file or included in the theme's JavaScript file.
- Resources:
 - <https://shopify.dev/docs/themes/ajax-api/reference/cart>