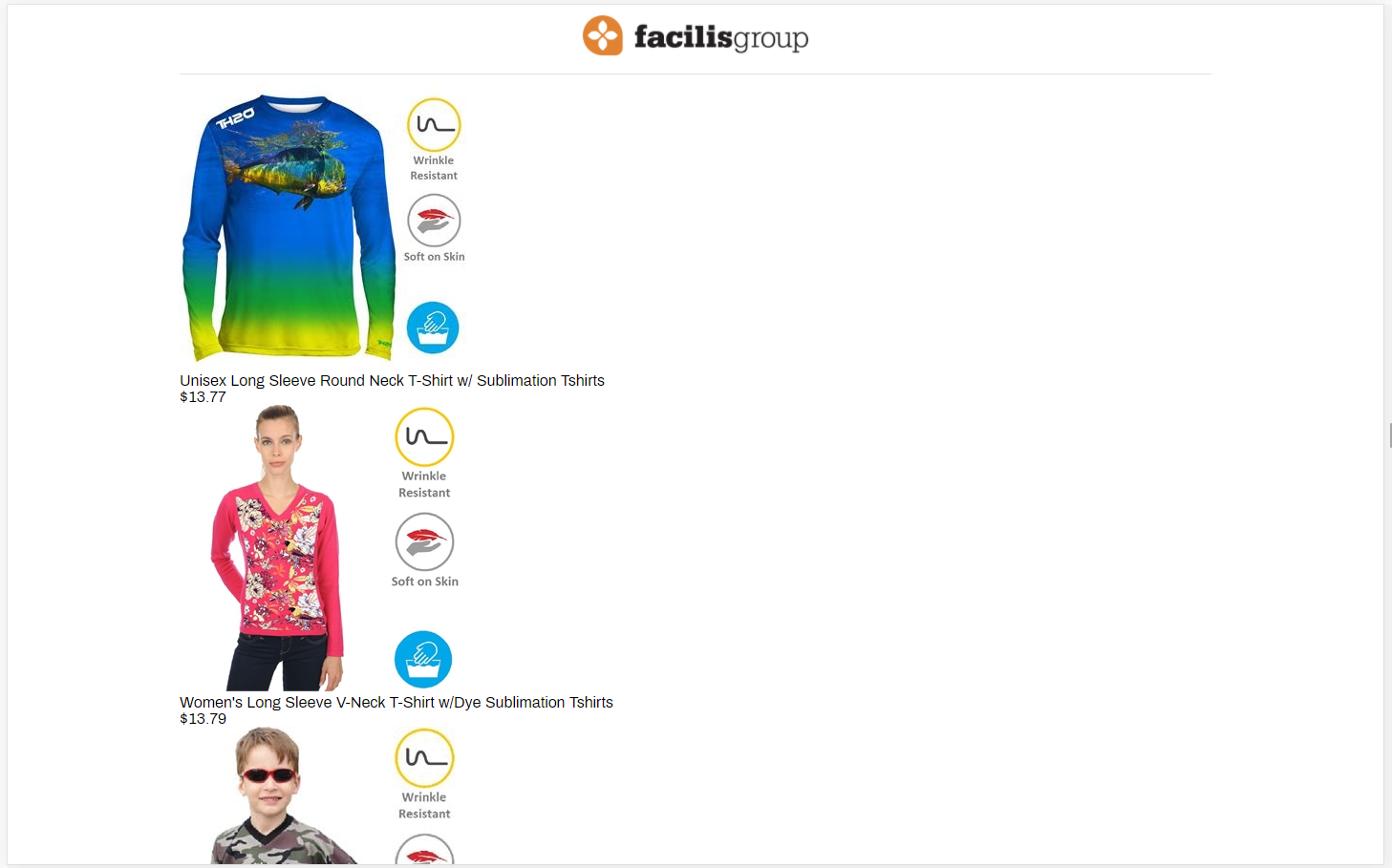
# Facilisgroup Commercio UI Technical Test

The goal of this test is to utilize the existing code and transform it in a functional UI where a user can page between a set of products. You can use any tools and frameworks (this goes for both JS and CSS) you need to complete the test. If you want to do it in vanilla JavaScript using nothing but DOM manipulation, ES6 and regular CSS, that is perfectly fine.

## The layout:

When opening the index.html file in the files folder, you will see a page that looks like this:

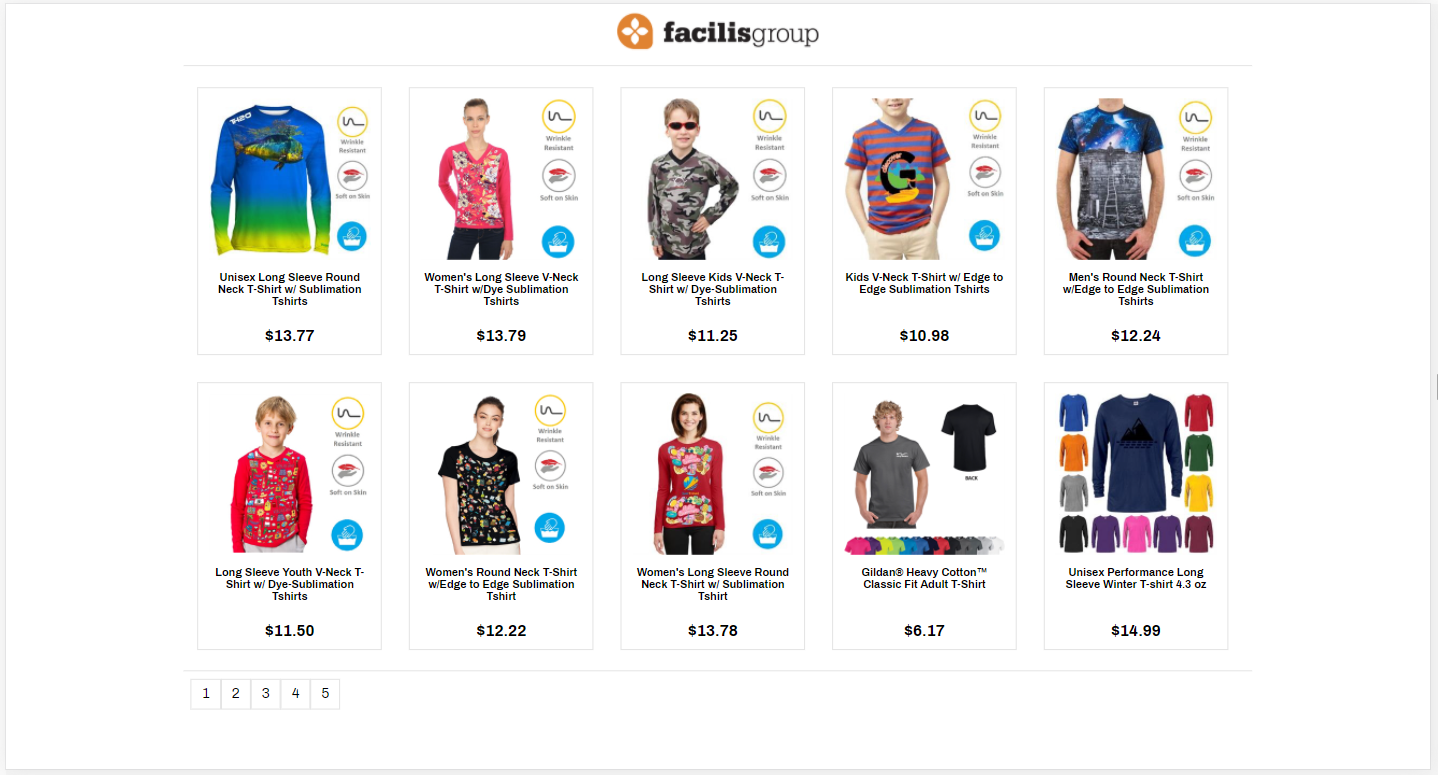


This page is currently missing any form of styling and the content is unfiltered. To complete this test, this page will need to be updated to ensure the user can do the following things:

* Able to see a set number of products (between 10 to 20) in a grid system.
* Able to click on a product and have its information displayed in a modal window
  + The information required to be displayed are:  
    **Product Name  
    Product Image  
    Product Description  
    Product Price  
    Supplier Name**
* Able to paginate between the products and see the result set of products.
* Full support for mobile

## The result:

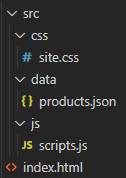
When completed, the UI could look something like this (on a desktop)



Important parts here are distinct visual separation between the products

## The files:

The project contains the following files:



The **site.css** file contains very minimal styling, mainly only for the body and header. You can make as many changes in here as needed.

The **products.json** is the required data source that will need to be used as a REST API endpoint.

The **scripts.js** is where the main JS that powers the app will need to be written. There are some placeholder functions written, but you are free to delete them and replace them with the logic needed to make this work.

The **index.html** currently has all products hard coded into the “products” element. This will need to be made dynamic and rendered through a function that is called after the products are loaded from the **products.json** from the **scripts.js** file. When done, there “products” element should have nothing inside when viewing the file.