**PRODUCT CATALOG v.9**

**Author: Terry Osterdock**

**Implementation** - <http://terryosterdock.com/projects/product_catalog/>

**Readme** - <https://github.com/terryo/product_catalog2/blob/master/README.md>

**Repo** - <https://github.com/terryo/product_catalog2>

**New technologies learned** – Backbone, JQuery, github. All completed with no assistance (apart from the internet)

**Overall Structure**

* Backbone markup (contents populated later)
  + Catalog
  + Product (for Add/Edit form)
* Router - funnels catalog and product (add/edit) views, passes in ID for edit, using display:none
* Model – data structure of a product
* Collection – catches updates to models that it contains and triggers event to update the DOM
* View
  + Setup callbacks for user actions/events (add, edit, delete, form submit)
  + add/editProductForm() – initializes the forms
  + add/edit/deleteProduct() – updates local storage and model/collection
  + setProductNode() – generates the markup for a new or edited product and adds/updates DOM
  + removeProductNode() – removes product from DOM
  + Lib/util functions
    - clearForm() – clears all form fields
    - getProductDetails() – grabs all values from product form and returns a model ready hash
    - addDummyProduct() – just adds an initial product on new user’s first load

**Included Libraries** – JQuery, Backbone, Underscore

**References used**

backbone api/overview - <http://backbonejs.org/>

model/collection/view overviews - <http://backbonetutorials.com/>

general questions - stackoverflow

router example - <http://aaronhardy.com/javascript/javascript-architecture-backbone-js-routers/>

design ideas - codepen.io file structure - <https://github.com/VIISON/BackboneSwitchView>

jquery api reference - <http://api.jquery.com/>

underscore api reference - <http://underscorejs.org/>

**A Few Features**

* Local storage indexes match product IDs to streamline insert/edit/delete
* Markup - Product\_elements array – utilized to populate form fields and create new product markup. This should make it easier to add a new field to a product (eg. Ratings, availability, etc)
* Initially tested for post, put, delete and json + commands should work with a restfull server when models get updated
* Currency ($), dimensions (inch/lbs) tags are populated via CSS so a different style could be applied per country.
* CSS – fairly decent color scheme, border radius, box shadows, gradients, transitions, transforms
* Commented and documented

**Feature/Task list (prioritized)**

* BUG - Fix Firefox (various errors) and fully cross browser test
* Split JS files into separate files (Router, Models, Views, Collection). Separate Catalog and Form views?
* Update to latest backbone & underscore libraries (latest listed in readme)
* BUG – going directly to a product edit form bypasses the editForm setup call and header text doesn’t get set and form doesn’t get prepulated with data - <http://terryosterdock.com/projects/product_catalog_dev/#product/1>
* Work with a designer to improve styling (match shipwire.com styling?)
* Form validation
* Styling - SASSify CSS
* Hook up a restfull server (should work fairly automatically via the model updates)

**Questions**

* Over-relying on JQuery? Could use backbone/underscore libs instead in some cases?
* Placement of product elements as a global within sandbox ok?
* Normal to have so much logic in the ‘View’?
* How should we approach templating?
* Why would changing certain body/section tags in DOM cause duplicate events to be fired?