



Msbith Task

- Delivery Products -

Task Definition:

You are tasked with creating a simple product management app.

The app displays the store's list of products.

The user can view the product list.

- Implement Server functionality that will receive the product list and validate its content.
- Build Html template using the provided UI Mockup below.
- The code needs to be coherent with meaningful class names, reusable styles, short code, ect...
- The page will have a responsive design need to support width of 300px to 1920px

Please send the link to Git with the Code to: gilad@msbitsoftware.com

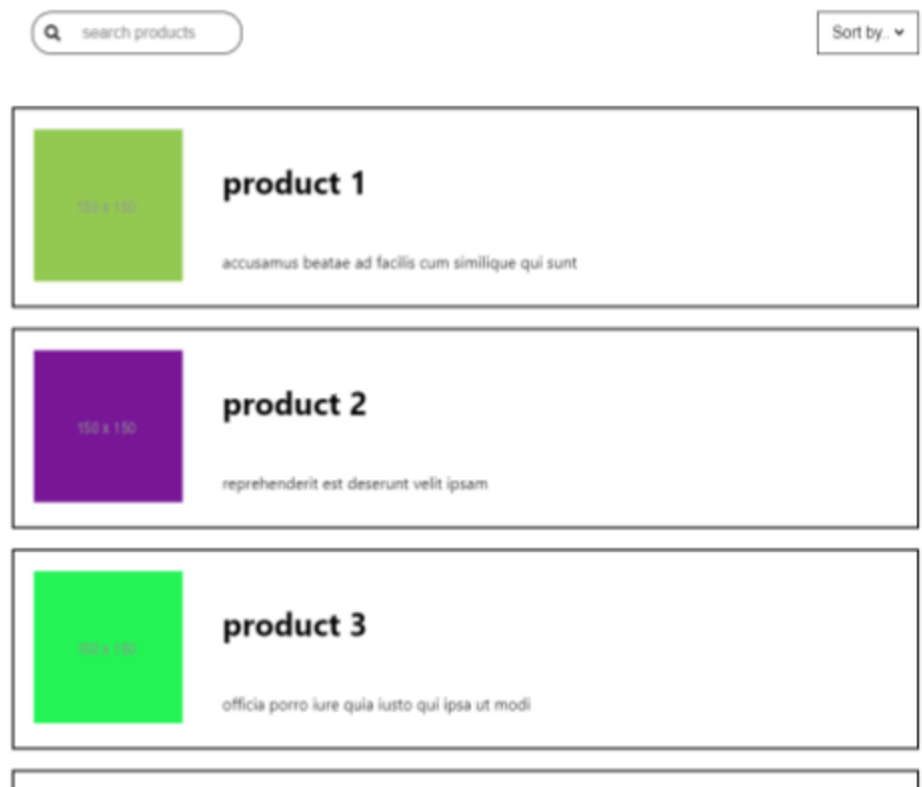
Overview:

Wire Frame (UI):

Mobile Layout



Desktop Layout



Entity Details:

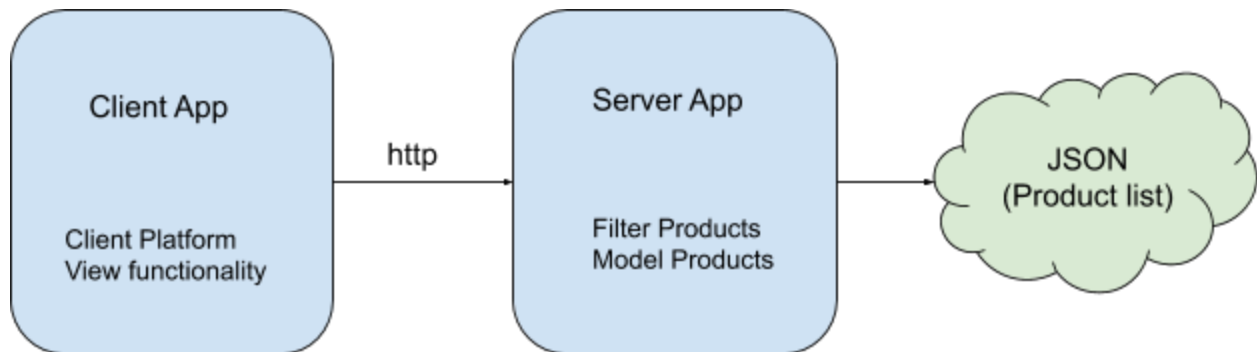
[Products.json](#) contains a list of products that are being delivered using several types of shipment services (UPS, Fedex, ect...).

Each Shipment service has its own status and type, and it may contain several products in one shipment.

Each Product has its own unique Id, Name, Description, Thumbnail Image, etc...

Instructions

Please build the app according to the following flow:



1. Server App: Data Access layer

Implement a Data Access layer service where you need to acquire a products data using an Api Call:

When Submitting an “getAll” api call:

successful call:

Request URL:

<https://msbit-exam-products-store.firebaseio.com/deliveryProducts/products.json>

Request Method: GET

Request Header: accept: application/json; charset=UTF-8

2. Server App: Business layer

Implement a business layer service where you will validate your input data and build a proper Model to be consumed in the client app:

Validation on data: only Items with shipment Status == 1 are valid (can pass on to client application)

Response Model: only required data needs to pass on to the client application.

The Params are: product id, name, description and image-thumbnailUrl.

**** Please note,** Product.Json files have 3 types of shipping services. Each type may contain one or many products. You will need to extract all of the product's data regardless of the type.

3. Server App: Controller layer

Implement a proper controller for the Client app to be able to invoke an api call.

The Api should be:

Path: .../api/products

Method: GET

Request Header: accept: application/json; charset=UTF-8

4. Client App:

Build a Client application that will display each of the products according to the WireFrame (need to be responsive).

Bonus:

1. Add the ability to sort the list by product name or price, by using a dropdown control.
2. Add the ability to search the list by free text into a text box control. The list should display only products whose name or description contain the search text.
3. Add 3D animation for each of the list images - use <https://github.com/micku7zu/vanilla-tilt.js> as the source for this functionality