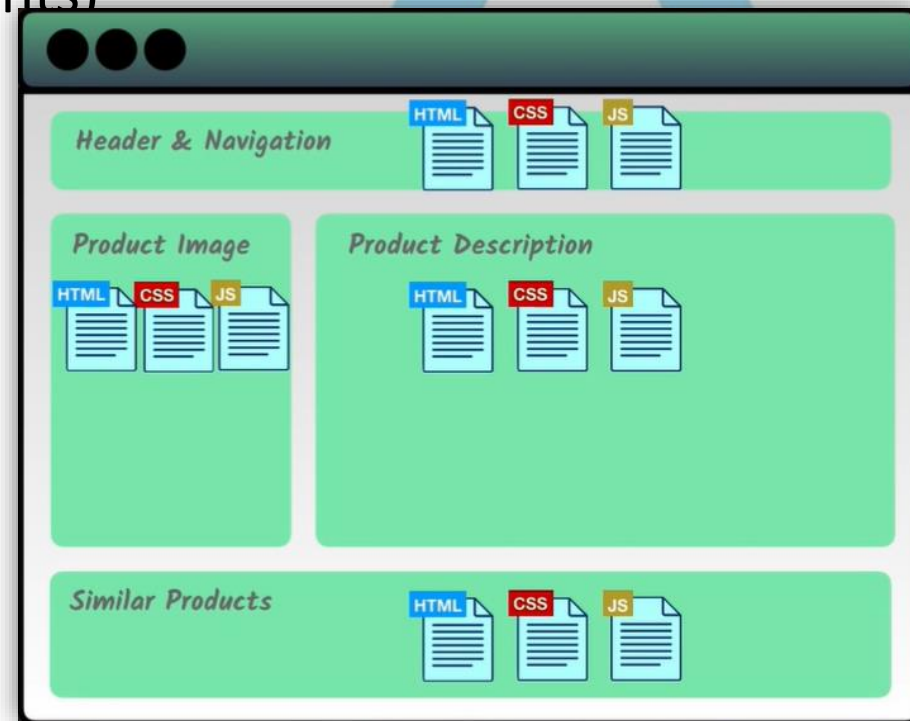
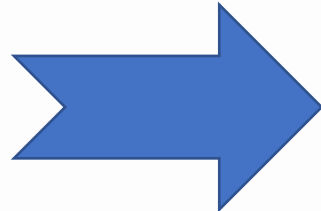
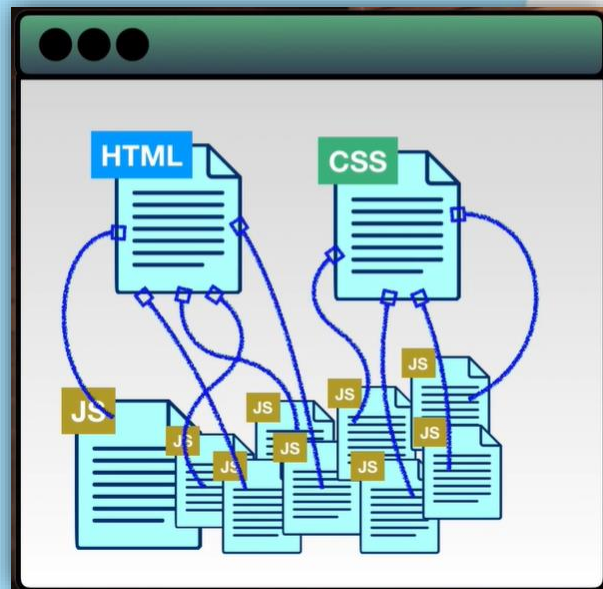


Module 3 Day 11

Introduction to Vue

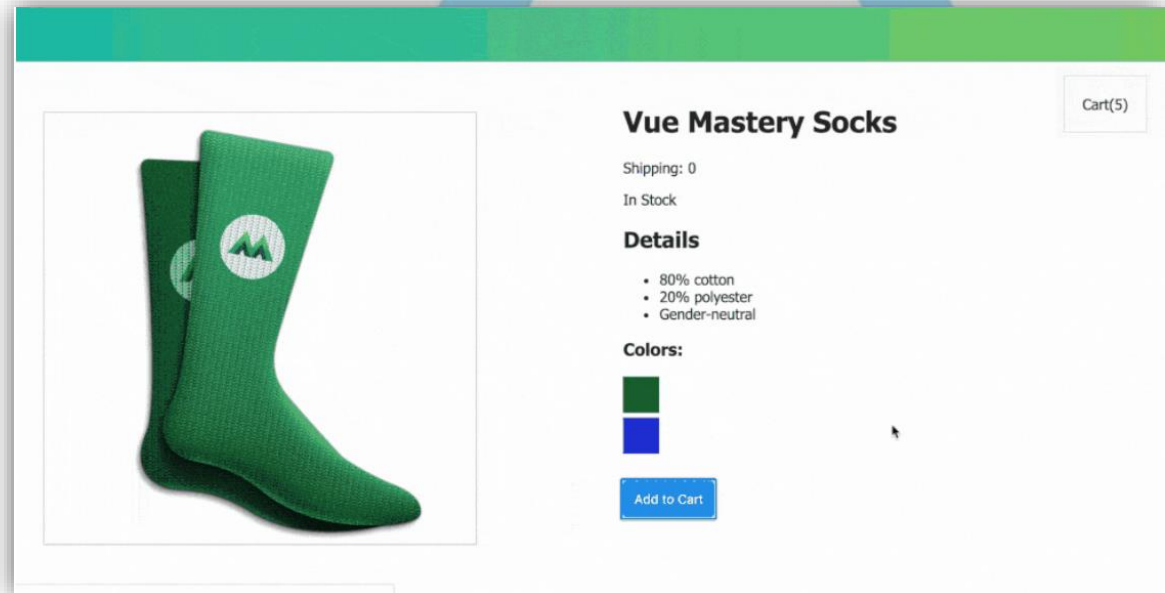
Why a Framework?

- Vanilla JavaScript is Powerful! (oh, and **TEDIOUS**)
- Builds applications in manageable chunks (components)
 - Built to stand alone and be testable
 - Placed into pages (or other components)
 - Contain HTML, JS and CSS



Vue

- Approachable, progressive
 - You don't have to rebuild entire apps; you can add Vue in as needed
- Very good documentation: <https://vuejs.org/v2/guide/>...
- And a video course library: <https://www.vuemastery.com/courses/intro-to-vue-js/vue-instance>...
- And...socks!
- <https://vuejs.org/v2/guide/#>



Preparation

Tool	Verify	Install
Nodejs and npm – Node and Node Package Manager. Node runs JavaScript files directly; npm manages add-on packages	> npm –version	https://nodejs.org/en/download/
Vue CLI – command-line interface for setting up Vue projects	> vue –V	> npm install -g @vue/cli
Vue.js dev tools for Chrome	chrome://extensions/ in Chrome's address bar. Look for "Vue.js devtools"	https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjimejigliipccpnannhbledajbpd
Vetur VSCode extension – syntax highlighting and IntelliSense for Vue.	In VSCode, Extension tab (ctrl-shift-X). Look for Vetur under Enabled.	In same tab, search for Vetur and click Install.

Creating a Project with Vue CLI

- > `vue create my-first-vue`
 - 140 MB of code!?
 - (139 MB of that is node_modules)
- > `code my-first-vue` (or launch VS Code from Explorer menu)
- > `npm install` (downloads all dependencies)
- > `npm run serve` (starts a local server and launches the site)
 - See package.json for its definition



Let's
Code

Project Structure

- node_modules – dependencies
- public – public website assets
- src/assets – website assets
- src/components – your component source files
- package.json – configuration / dependencies

Vue Components

- A .vue file defines a component in Vue
- Display (HTML): **<template>**
 - <template> must contain a single child element
- Style (CSS): **<style scoped>**
 - <style scoped> tag allows CSS to apply to only this component
- Data and behavior (js): **<script>**
 - Properties and functions on the object provide data and information about the object
 - *data* is a function that returns an object. <https://alligator.io/vuejs/component-instancing/>
- App.vue – the "root" Vue instance

JavaScript Objects Review

- Objects are keys and values
- Properties can be any data type:
 - Strings, numbers
 - Arrays
 - Objects
 - Functions

```
let rectangle = {  
  length: 20,  
  width: 34,  
  area() {  
    return this.length * this.width;  
  },  
  perimeter: function() {  
    return (2 * this.length) + (2 * this.width);  
  }  
};  
let totalArea = rectangle.area(); // Will equal 680  
rectangle.length = 49;  
rectangle.width = 82;  
totalArea += rectangle.area(); // Will equal 4698  
console.log(rectangle.perimeter());
```

Let's
Code

Vue Components – JS Properties

- `data()`
 - Function that returns an object that contains the component's "properties"
- `computed`:
 - An object full of functions that return computed values and then are treated as property getters on the component
- `props`:
 - An object with properties for data that is passed in from the "parent" component when created
- `methods`:
 - An object full of functions to be treated as methods on the component

Data Binding

- Linking a data property of a component to a specific HTML element
- One-way (Simple): Component data is viewed in the UI {{ }}
- Two-way: Component data is viewed and updated from the UI
 - v-model
- v-model modifiers (<https://vuejs.org/v2/guide/forms.html#Modifiers>)
 - .lazy – update after focus is lost
 - .number – assign the property as a number instead of string
 - .trim – trim whitespace from before and after a string input



Demo

Loops (v-for)

- Arrays
 - v-for="item in items" -- equivalent to C# "foreach(Item item in items)"
 - v-for="(item, index) in items"
 - v-bind:key="uniqueValue"
- Objects
 - v-for="value in object"
 - v-for="(value, key) in object"
- Integer
 - v-for="i in n" – eq. to c# "for (int i = 1; i <= n; i++)"
 - Sets n = 1 to integer in a loop
 - That's right, it starts at 1!



Let's
Code

Binding DOM Attributes (v-bind)

<https://vuejs.org/v2/api/#v-bind>

- `v-bind:class = {
 'class1' : Boolean-expression,
 'class2' : Boolean-expression
}`
- `v-bind:style = {
 'color' : barColor,
 'width' : barValue + 'px'
}`
- `v-bind:disabled="Boolean-expression"`

Let's
Code