Electric Ladyland

https://essayerdeux.herokuapp.com

MEAN (

Mongo

Express

AngularJS(Angular)

Node (node.js)

)

Project ‘Electric Ladyland’ is single home page of a marijuana dispensary who has a bit more of a design purpose. Using, Mongo, Express, Angular and Node, I created a dispensary that aimed to answer one of MANY important topics on the table in the marijuana industry. That topic is, how to protect new marijuana users who don’t know how intoxicated they will get? Maureen Dowd, in the NY Times article “Don’t harsh our Mellow, Dude”, writes about her experience eating a too much of a cannabis edible too fast.

Dowd writes, “What could go wrong with a bite or two? Everything, as it turned out. Not at first. For an hour, I felt nothing. I figured I’d order dinner from room service and return to my more mundane drugs of choice, chardonnay and mediocre-movies-on-demand. But then I felt a scary shudder go through my body and brain.

I barely made it from the desk to the bed, where I lay curled up in a hallucinatory state for the next eight hours. I was thirsty but couldn’t move to get water. Or even turn off the lights. I was panting and paranoid, sure that when the room-service waiter knocked and I didn’t answer, he’d call the police and have me arrested for being unable to handle my candy.”

As an Otter cohort member, we were encouraged by Rafa to “think about an app that solves a problem”. So, I thought about how to conveniently, relationally and practically inform new users of cannabis just how “How high would I get?”. Or as I like to say, “How wasted?”, “How hung over?” or “How intoxicated?”.

#G.R.O.T. General Rule of Thumb

Weed (a.k.a.: Cannabis, Marijuana) & Alcohol (a.k.a.: happy hour, beer, chasers) impair motor skills, judgment and all cause the ‘hung over effect’. This project attempts to answer a problem, how to best inform new users of cannabis just how intoxicating 1 gram of marijuana leaf will be or half a bar of a marijuana edible. Use alcohol, something MANY adults use and have a frame of reference to use to evaluate just how intoxicated they are or can get after 2 beers in an hour, or 3 margaritas at a happy hour event or beer and pizza at a hack-a-thon.

More from Maureen Dowd’s article, “Don’t harsh our Mellow, Dude”,

“Andrew Freedman, the state’s director of marijuana coordination. “There are way too many stories of people not understanding how much they’re eating. With liquor, people understand what they’re getting themselves into. But that doesn’t exist right now for edibles for new users in the market. It would behoove the industry to create a more pleasant experience for people.”

Articles:

* NY Times: http://www.nytimes.com/2014/06/04/opinion/dowd-dont-harsh-our-mellow-dude.html?\_r=2
* Dank : http://www.dank-colorado.com/product/chem-dawg-4/
  + The inspiration on this specific site is the ‘Used to help’ section which displays percentages of “Depression”, “Stress” and “Pain”.

MEAN Technology

Mongo Database:

Using my node package, mongoose is required and developed here in the ‘server’ directory and inside the ‘models’ subdirectory. The name, description and rating are used to display the marijuana leaf (it’s name, description of the flower and the THC % value of the leaf).

var mongoose = require('mongoose');

var productSchema = mongoose.Schema({

//strain

name: { type: String, required: true },

description: { type: String },

rating: { type: String },

}, {timestamps: true});

//});

module.exports = mongoose.model('Product', productSchema);

//Models are fancy constructors compiled from our Schema definitions.

//Instances of these models represent documents which can be saved

//and retrieved from our database. All document creation and retrieval from the database

//is handled by these models.

In app.js, mongoose is required and connected to find a database on the site where I am hosting it or use the one I provided.

mongoose.connect(process.env.MONGOLAB\_URI || 'mongodb://localhost/products\_api');

Angular

My ‘mainController’ performs two data binding to display the newly entered marijuana leaf.

$scope.updateProducts = function(){

productsApi.getAll().then(function(response){

$scope.products = response.data.products;

});

};

$scope.create = function(){

productsApi.create( $scope.newProduct ).then(function(){

$scope.updateProducts();

});

};

$scope.delete = function(id){

productsApi.delete( id ).then(function(){

$scope.updateProducts();

});

};

My ‘productsInterface’, works with ‘mainController’ in that it gives ‘mainController’ the data/information. This step comes right before displaying the newly entered data.

productsInterface.getAll = function(){

return $http.get( baseUrl );

};

//think databinding ... example <form ng-controller/ ng-model="newProduct.name/.description/.rating/.img"> </form>

productsInterface.create = function( newProduct ){

var payload = { product: newProduct};

return $http.post( baseUrl, payload);

};

productsInterface.delete = function( id ){

return $http.delete( baseUrl + id );

};

Using Angular directives, my information entered in the below form input fields will be displayed onto the homepage site of the application.

<div class="row">

<form id="new-entry" ng-submit="create()" class="col s6">

<input class="black-text" type="text" ng-model="newProduct.name" value="" placeholder="...strain">

<input class="black-text" type="text" ng-model="newProduct.description" value="" placeholder="...description">

<input class="black-text" type="text" ng-model="newProduct.rating" value="" placeholder="...rating">

<button type="submit" name="button">Save</button>

</form>

</div>

Thanks to two way data binding, my views are instantly updated.

<ul>

<li id="product-lists" ng-repeat="product in products">

<h3>{{ product.name}}</h3> <br><p>{{product.description}}</p> <br><h6 id="thc" value="">THC: {{product.rating}}

<br><a href="#" ng-click="delete(product.\_id)">Delete</a>

</li>

</ul>

Thanks to a little bit of JavaScript, users can hover over images of beer, tequila shot glass and a wine glass, and get a conversion of that marijuana’s leaf THC value to a specific amount of alcohol in an hour.

<div class="display-wine" >

<h6>one gram is like 3 glasses of red wine in 90 min... <br /><br /> #GROT: wine has more alcohol than beer <p id="exit-wine">X</p></h6>

</div>

function displayWine(){

$(".wine").hover(function(){

$(".display-wine").css("visibility", "visible");

});

}

function hideDisplayWine(){

$('#exit-wine').click(function(){

$('.display-wine').css("visibility", "hidden");

});

}

.wine{

width: 4%;

height: 3%;

position: absolute;

top: 17%;

left: 80%;

}