**Portfolio Boiler Plate**

**Members**: Timothy Rogers, Ben Holmes

**Description**: An AngularJS Portfolio site template that can be easily adapted to specific users.

Table of Contents

CODING 3

SNIPPET 1 3

SNIPPET 1 A 3

SNIPPET 1 B 3

SNIPPET 2 3

SNIPPET 2 A 4

SNIPPET 3 4

SNIPPET 3 A 4

SNIPPET 4 4

SNIPPET 4 A 5

SNIPPET 4 B 5

SNIPPET 4 C 5

Overview Page 6

New and Complex Issues 6

Work Towards Bloom’s Taxonomy 7

New And Complex Section 9

New Things 9

Complex Things 10

Bloom’s Taxonomy Section 11

# 

# CODING

## SNIPPET 1

.**fade1**.**ng-enter** {   
 **animation**: **scaleUp** 0.5**s both ease-in**;   
}  
.**fade1**.**ng-enter-stagger** {  
 **-webkit-transition-delay**: 0.3**s**;  
 **transition-delay**: 0.3**s**;  
 **animation-delay**: 0.4**s**;  
}  
.**fade1**.**ng-enter-active** {   
 **opacity**: 1;   
}

#### SNIPPET 1 A.)

<**div class = "container mb2"**>  
 <**div ng-if = "show" class = "fade1 text-blue" ng-repeat = "item in vm.items track by $index"**>  
 <**div class = "text-center mt15"**>  
 <**h4 class = "ml5 textTitlePortfolio"**>{{item.name}}</**h4**>  
 <**a ui-sref={{item.href}}**><**img class = "circlePic" src = {{item.img}}**></**a**>  
 </**div**>  
 </**div**>  
</**div**>

#### SNIPPET 1 B.)

$scope.$watch(**"show"**, **function**() {  
 **var** url = **window**.**location**.**href**;   
 **if**(url.indexOf(**"portfolio"**) != -1){  
 $scope.changeShow();   
 }  
 **else**{  
 $scope.**show** = **false**;   
 }  
});

## SNIPPET 2

angular.module(**'myApp'**)  
.filter(**'dayFilter'**, **function**(){  
 **return function**(data, scope) {  
 **var** result = [];   
 **for** (**var** i =0; i < data.**length**; i++) {  
 **if**(data[i].**id** == scope.vm.**daySchedule**)   
 {   
 result.push({**id**: data[i].**id**, **name**: data[i].**name**, **status**: data[i].**status**});   
 }  
 }   
 scope.vm.**theseTasks** = result.**length**;   
   
 **return** data.filter(**function**(element, index, array) {  
 **return** element.**id** == scope.vm.**daySchedule**;  
 });  
  
 }  
   
});

#### SNIPPET 2 A.)

<**li data-ng-repeat="task in vm.tasks | dayFilter:this track by $index"**>  
 <**input ng-if="false" type="checkbox" data-ng-model="task.status"**> {{task.name}}   
 <**span class="todoSpan glyphicon glyphicon-remove" ng-click="vm.removeTask($index)"**></**span**>  
</**li**>

## SNIPPET 3

**for**(**var** i = 0; i < 100; i++) {  
 pxs[i] = **new** *Circle*();  
 pxs[i].reset();  
}  
setInterval(*draw*,rint);

**function** *draw*() {  
 con.clearRect(0,0,WIDTH,HEIGHT);  
 **for**(**var** i = 0; i < pxs.**length**; i++) {  
 pxs[i].fade();  
 pxs[i].move();  
 pxs[i].draw();  
 }  
}

#### SNIPPET 3 A.)

<**canvas id='floating'**> </**canvas**>

## SNIPPET 4

<**div class = "row text"**>  
 <**div class = "col-lg-5 col-lg-offset-1 col-md-5 col-md-offset-1 col-sm-5 col-sm-offset-1 col-xs-5 col-xs-offset-1"**>  
 <**p class = "text-black mt5"**>Best Actor (2016 The Martian)</**p**>  
 <**p class = "text-black mt5"**>Best ScreenPlay (1998 Good Will Hunting)</**p**>  
 <**p class = "text-black mt5"**>Best Supporting Actor (2010 Invictus) </**p**>  
 </**div**>  
 <**div class = "col-lg-4 col-lg-offset-2 col-md-4 col-md-offset-2 col-sm-4 col-sm-offset-2 col-xs-4 col-xs-offset-2"**>  
 <**p class = "text-black mt5"**>Best Cast Ensemble (2006 The Departed) </**p**>  
 <**p class = "text-black mt5"**>Outstanding Artistic Contribution (2007 The Good Shepherd) </**p**>  
 <**p class = "text-black mt5"**>Favorite Actor (2000 The Talented Mr. Ripley) </**p**>  
   
 </**div**>  
 <**hr**>  
</**div**>

#### SNIPPET 4 A.)

**font-size**: 1**em**;

#### SNIPPET 4 B.)

**width**: 100%;

#### SNIPPET 4 C.)

**border-top**: 1**px solid #777**;

#### 

# Overview Page

## New and Complex Issues

ngAnimate is one of the really interesting things we learned. The concept is really cool because the ngAnimate injection adds extra classes, to handle the entrance and exit of a div element. So we had to learn about the ways to handle these entrances. Snippet 1 (above) shows an example of how we did this.

There was an additional obstacle with using ngAnimate on the Portfolio page. We had an error, where the item animation would only appear on the first visit to the Portfolio page, and every visit after that would open the Portfolio page with the items already showing, without being animated into motion. To fix this we created an ng-if statement and use a Boolean ‘show’ as its condition. This is shown in Snippet1 A. The ‘show’ variable is always false when you are not on the Portfolio Page, and it is true when you are. We used a watch , which is shown in Snippet1 B.

Using a custom angular filter was another new obstacle with implementing the schedule from the Calendar page. We use an ng-repeat to show items that need to be finished, but the problem was filtering the completed from the uncompleted while also filtering which day is currently being selected. We saw this as a situation where we needed to filter by two different criteria, so we created a custom filter called ‘dayFilter’ . This is shown in Snippet 2. The html required to use this filter is in Snippet 2 A.

## Work Towards Bloom’s Taxonomy

-Creation

We wanted to build a Portfolio page that would look clean, be easy to use, and could be easily adapted for different people. We decided it should contain a Home page, About page, Portfolio page, and some sort of additional filler page, which we chose to show the Resumé.

We started with the Home page, and to add some visual appeal, we implemented the Canvas to show some floating star projectiles. The basic implementation of the canvas is in Snippet 3. We thought this would be a nice effect. The html used for the canvas is in Snippet 3 A.

The About page was built pretty simply, just using Bootstrap for the CSS, and creating a basic informational page. We tried to make the site relatively mobile friendly, so for our grid layouts, we provided all versions of column sizes, which can be seen in Snippet 4. We also used em to measure text, percentages for all size relations, and mainly hex values for colors. All these examples can also be seen in Snippets 4 A, B, C.

Creating the Resumé page turned out to be very simple, because we just converted a resume .pdf to a .png, and displayed it as an image on the page. The final page, the Portoflio took the most time, because we had to use the ngAnimate, and implement some type of Portfolio worthy material. We decided on a Calendar, Basic Javascript game, Google Maps API, and two space holders for 309 and 319.

-Evaluation

We definitely wanted whatever we did, to have some visual appeal, as well as expand and build on concepts we have learned about in 319. This was the main reason that led us to using ngAnimate (which is a new AngularJS feature to us), and the Canvas for the home page (which was an improvement on what we learned in 319).

We also wanted the layout to be resizeable, and easy to navigate. We used Bootstrap to create the resizeable navbar, and then we also used em text sizing, percentages for size relations, and bootstrap classes for the grid layout. Neither of us had ever created a completely mobile ready website before, so we though this would be a cool concept to focus on. Some research taught us that bootstrap is already mobile ready, if the right classes are used, and then that the use of em and percentages will help items stay in the correct relation to one another.

With these concepts in mind, we created the Portfolio site.

-Analysis

This site was created as a place for people to host their portfolio projects, and showcase their abilities. It is simple to navigate, resizable, and pretty easy to customize. The backgrounds and all the content can be changed, and adding more projects to the Portfolio is easy because all that needs to happen, is the URL of the new project needs to be specified in the PortfolioCtrl.

# New And Complex Section

## New Things

One of the most interesting new things was the ngAnimate, and making that work with the use of a timeout. Unfortunately we don’t think ngAnimate was any easier to use than the jQuery fadeIn() fadeout() functions, but we thought it would be important to learn effects the Angular way.

The filter was a great new thing to learn because it provides a new way of determing which items to display in an ng-repeat. It really opens a lot of possibilities when it comes to filtering the information because it basically just means you can filter over whatever criteria you want without having to modify the information in the array being repeated over.

Other new concepts that we became more aware of as we created this project was that page layout’s generally looked better when they weren’t crowded, or didn’t have too much going on. For this reason we tried to keep everything as simple as we could, easy to write, easy to read.

The Google Maps API was also a helpful to new thing to learn. We both wanted to do something with the Google Maps API, like we did on the first assignment, so we decided to make a basic implementation and add it to our portfolio. Most of the code included can be found in Google Developer sites, but we added our own twist to all the functionality.

## Complex Things

One of the hardest parts for us was just coming up with interesting designs for the pages, and trying to improve the user experience as much as possible. Neither of us really have a lot of experience with graphic design or art, so making a site like this look pretty turned out to be difficult. As stated above, we realized that the pages with the least amount of stuff going on tend to look the best. We used this principle to overcome our design difficulties.

Another complex thing, that just kind of fell into place was the implementation of Google Maps. We created a basic user interface for the maps, and it allows people to search by location type (restaurant, school, gym), and limit the distance by miles. It also identifies the user’s location, and automatically pin points it on the map, using this as the starting point. You can then save different locations you’d like to visit, click on them, and get the distance it takes to travel from A to B. Working with Google Maps is complex, because there is so much data to navigate, but Google Developer really makes it easy, and provides a lot of code implementation.

The final complex thing we would like to mention in a little more detail is the filter used in the calendar. It took us a while to understand how this was working, but it turns out, every time the ng-repeat iterates to the next item in the array, it calls the filter, and passes the filter the entire array, instead of just the object being iterated, so inside the filter we had to iterate the whole array, and return the items that match the specific criteria.

# Bloom’s Taxonomy Section

**-**Creation

In order to implement a visually pleasing, simple design, we decided to spend a lot of time searching through the bootstrap pages, and learning about the basic guidelines for writing our classes. We used these guidelines to create the nav bar, and basically all the CS in the site, with the exception of some custom classes. We created basic CSS classes that handle easy changes, like margins and padding. Otherwise we tired to follow the bootstrap standards for everything, including making the site mobile ready.

Another thing we did to prepare for creating our website was searching through some different angular sites, and trying to find some design inspiration. We just googled, “Cool Angular Sites” and looked through the templates. We found a lot of really nice looking designs, but instead of copying something exactly we decided to create something original. We used circular images with the idea of a mobile site in mind. We thought a circular image would be easy to click on because it approximates the shape of a thumb. We also thought white was the best background choice, just because it gives a little bit of a classier look. We both would like to add more color to the design, but we weren’t very inspired.

The ultimate guideline we used in our site creation was just trying to follow a clean folder structure, and using angular aspects. We implemented directives, controllers, filters, ngAnimate, and angular specific methods like Angular.copy and other things. Since this is the final project of the year, we were more interested in doing a recap so we could solidfy the concepts we have learned such as canvas, the google maps api, jquery, and angular.

**-**Analysis

The final product is cleanly designed, easy to use, and relatively mobile friendly, which is what we wanted. By relatively friendly we mean there are parts that can be improved, but everything does scale correctly, which you can see by simply resizing the window while the site is open. The one exception is the resume page, which doesn’t fit the .png photo for smaller devices, and the fact that sometimes the writing gets too small to read on a mobile phone.

We are proud of the work we did with the Google Maps API, and especially with finding the user’s location. We are also proud to say that everything on the site does work, and nothing is broken. We did all our development using Google Chrome, so there is a possibility that some parts of the website won’t work for other browser’s such as safari or Mozilla.

We are also thankful for the opportunity to learn about the filter directive, which looks like it could be very useful in the future. It provides of quickly searching through the values in an array, without ever having to actually change anything in the area. This is a little quicker than modifying the array, because for the implementation you just need to add the filter to the ng-repeat, and then include the JS file for the filter in the index.html.

Upon looking at the finished design, we think there is room to add a little more color, and flavour to the layout, but that may be a project for another time. As far as what we initially set out to accomplish, we did meet all our goals.

**-**Evaluation

We decided to design our app the way we did because of a few reasons, most of which have been covered above. We wanted to reinforce what we have already learned in the class, while also improving on their concepts. This led us to design the application using AngularJS, and also to implement the canvas and google maps in the portfolio section.

We also wanted our folder structure to be easy to navigate, and this was managed using a lot of what we learned from Portfolio 2 about the mvc structure. We have folder directories for every page in the app, and each of those directories contain views, controllers, directives, services, or whatever is associated with that page. We also decided to use the constants directive again, and to store all the images associated with the site inside a directory labelled images.

Beside basic ease of use, readability, and design, we wanted the content to be sufficient for our final Portfolio. We built on concepts we have already learned, while also implementing new parts of AngularJS that we have never used yet, such as ngAnimate, and filter. We both are starting to feel pretty confident in our AngularJS abilities.