Pass Task 6

Model-view-controller programming is a software design approach that is used in creating and designing web pages. Model is basically the logic or knowledge. View is the representation of the model visually where it can update the model when there is a change and vice versa. Controller connects both the model and the view. It provides menu or other user outputs. The model passes off the information to the controller and the controller outputs messages that the user understand to the views. Bootstrap 3 is mobile first. For bootstrap, the grid system is used to properly layout the items on the html for both desktop and mobile view using a 12 column system. This is to ensure responsiveness for both desktop and mobile as the columns will rearrange itself according to the screen size. Col-xs is for small screen, col-sm is for tablet, col-md is for desktop while col-lg is for large desktop. Based on the credit task 6, I learn how to properly layout the columns using row.column to first draw out a template which conveys the structure of a bootstrap. The rol and col would be called in the div class such as   . A is used to contained the entire rol and col. Then, I implemented it into my website by referencing the Vincero watches website. By trying to replicate the website, I learnt how to properly layout my items in both desktop and mobile so that my website is more responsive to the changes in screen size. This helps me understand how the columns work and its default properties such as the default margin and padding and how I can manipulate the columns using both css and the bootstrap properties or content such as img-responsvie so that my images are responsive to the changes to the columns and many more. Another to note is the bootstrap contents, which by referencing the web app to bootstrap.min.js, bootstrap.min.css and many more, you could access the bootstrap properties. For example, the default h1 for bootstrap 3 is set to 36 px and uses font Helvetica Neue. What this means is the bootstrap items is already globally styled. For example in pass task 1, we are required to include in the required bootstrap files and then told to write hello world using h1. We can tell visually that it is already styled for us. For form controls in bootstrap, .form-control class gives us textual elements such as <input>, <select> and <textarea> with default width of 100% so it scales with the screen size. For example this can be seen in our register.html, where the registration

form scales with the screen size so the items do not stacked against each other when screen becomes smaller. For AngularJS, it is javascript framework comes with declaration templates and dependency injection. We use angular.module as it is a collection of functions such as services, directives, controllers, filters, and configuration. The module can be declared in html to define the start of the module using data-ng-app=”myApp” which can be seen in all the tasks involving the use of angular. We also use controller to bind a specific view in the html and to manipulate the view. The start of a controller starts with declaration of data-ng-controller=”myCtrl”. Ng items declared in that controller will only be manipulated in the same controller. This could be seen in the tasks units.html where we ng-init an array of units and its details and then ng-repeat them into the table. These ng items could only be used in the same controller as we cannot repeat those arrays into another controller or view. Besides, scope binds html and javascript and acts as a data model. $scope can store value or act as a function in a controller which can then be rendered in the html using {{*expression*}}. This can be seen in the guessNum.html where we are asked to input our guess. Our input is bind to a scope and then when we ng-click a button, it runs the $scope.function to calculate our $scope.guess against the orginal to see if our guess is correct and if it is not how far are we from the answer which is rendered messages into the html using $scope to store the messages string which activates based on certain criteria. For filter, to use filter we need to use a pipeline | to filter out a subset of an array and return it as a filtered array based on the conditions. Eg. In the units.html we are using our inputs as filters which specific units that we want and return them instead. Custom filter is used to format certain inputs such as converting number into roman numerals. Directive is used to reduce the amount of codes we need to write if there is a particular code that repeats itself as it basically acts as template that can be reused. In the phone product html, the structure of the code are the same for all products so we just declared the same template in our directives and just reused them to render the same structure for four different products. Ng route, basically route a another view into our index.html. In the routing for units.html, I basically route the view that returns the detail of a course I clicked using ng-show from another html. Angular also carries out client-side form validation and monitors input and form states. An example would be, $valid is which form is valid and $dirty in which something had been modified.

References

AngularJS. 2019. *What are Scopes?*. [ONLINE] Available at: https://docs.angularjs.org/guide/scope. [Accessed 5 May 2019].

Angular.io. 2019. *Form Validation*. [ONLINE] Available at: https://angular.io/guide/form-validation. [Accessed 5 May 2019].