**HAFTALIK İÇERİK DETAY**

|  |  |
| --- | --- |
| **Yapılan İş:**  Şirket Tanıtımı, Tanışma | **Tarih: 25.05.2015** |
| 2005 yılında kurulan P.I.Works, network optimizasyonu çözümleri ile dünya çapında mobil network operatörlerine hizmet vermektedir. Bugün 13 ülkede, 18 operatör ile çalışan P.I.Works, yenilikçi ve güvenilir optimizasyon çözümlerini sunarak, milyonlarca kullanıcıya kesintisiz iletişim imkanı sağlamaktadır. Bunun yanı sıra sunduğu teknoloji ile mobil network operatörlerine abone başı senelik ortalama 4 USD’lık bir tasarruf sağlamaktadır. Bu, operatörün müşteri hacmine ve teknolojiyi kullanabilme potansiyeline göre yıllık on milyonlarca USD tasarruf anlamına gelmektedir.  Kurulduğu günden bu yana her yıl ortalama %100 karlılık sağlayan P.I.Works 2010 yılında %264 büyüme kaydetmiştir. Önümüzdeki 3 yıl içindeki hedefi cirosunu %1000 oranında artırmak olan şirketin uzun vadeli hedefi ise Nasdaq’a kote olmaktır.  Tanışma: Şirkette staj yapacağım yer gösterildikten sonra, şirket tarafından sağlanan bilgisayara ilk ayarlar yüklendi, şirket mailim hazırlandı, staj sırasında kullanacağım programlar ve dökümanları yükledim. Staj boyunca bana rehberlik edecek Altay beyle tanıştık. Stajımda benden beklentilerini ve yapmam gereken şeyleri gösterdiler.  Daha sonra staj boyunca giriş çıkışlarda kullanmam içimn parmak izi okuyucu , giriş kartı ve yemek kartını aldım. | |
| **Yapılan İş:** Görev tanımı ve angular’a giriş | **Tarih: 26.05.2015** |
| Stajımda benden istenenler tanıtıldı. Aşağıdaki dinamik takvim kullanılan bir sistemin bir kısmını göstermekte, bu takvim istenilen herhangi iki tarih arasını saat bazında rendelemekte, bu saatlerden istenilen aralığın herhangi saatleri yada aralıktaki belli bir iki yer seçilip daha sonra bu seçilen yerler de gösteren bir rapor hazırlanabilmekte. Dinamik takvimde tarih aralığı genişledikçe rendeleme olayı uzamakta buda bir aralıktan sonra biraz performans düşüklüğü yaşamakta.  Staj boyuncaki görevim bu dinamik takvimin aynısını temel olarak angular js i kullanarak daha hızlı hale getirmekti. Bunun için ilk olarak angular jsi öğrenmem gerekiyordu.  C:\Users\hello\Desktop\piworksworkplace\piworks\appscreanshoot.jpg | |
| **Yapılan İş:**  Angular tanıma ve ilk kodlamalar | **Tarih: 27.05.2015** |
| Staj boyunca çalışma ortamı olarak visual stüdyoyu dokuman olarak pro angular js kitabını ayrıca https://**angularjs**.org/ sitesini çokça kullandım.  AngularJS, google’nin desteklediği, javascript MVC kütüphanesidir.  MVC (Modal View Controller) verinin, görünümün ve kontrollerin ayrı tutulmasına dayanır.  Modal: Verilerin tutulduğu nesnelerdir. İlişkili olduğu View nesnesini tetikler.  View: Görünümün (template) tutulduğu nesnelerdir.  Controller: Modal ve View nesnelerinin bağlandığı nesnedir. İşlemler ve kontroller burada yapılır.  Anguların avantajları: AngularJS çalışmak için jQuery gibi kütüphanelere ihtiyaç duymaz. Tek sayfa uygulamaları geliştirmeye olanak sağlar. Bu tarz diğer kütüphanelerden en önemli farkı çift yönlü olmasıdır. Yani veride olan değişiklik görünüme, görünümde olan değişiklik de veriye uygulanır. [Directive](http://harunozer.com/makale/angularjs_custom_directive.htm) tanımlayarak html etiketlerini geliştirilmesine de olanak sağlar. Kodların kısalmasını sağlar kod karmaşasını büyük ölçüde önler.  Anguların birçok kütüphanesi ve servisi vardır. Staj boyunca çoğuna ihtiyaç duydum, yapacağım programda çok kritik öneme sahiptiler. | |
| **Yapılan İş:** Angularım mimarisini anlama | **Tarih: 28.05.2015** |
| Ilk olarak anguların javascript dosyasını tanıtmak gerekiyor, bunun için head tagları arasına direk linki verebiliyoruz yada angular js dosyasını indirip olduğu dizini de head tagları arasına yazabiliriz, internet gittiğinde sorun yaşanmaz o şekilde.  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" ng-app>  <head>  <title></title>  <script src="http://code.angularjs.org/snapshot/angular.js"></script>  </head>  Angulardaki kodların çalışması için buddy tagın ng-app yasmak gerekiyor yoksa html kodlarımızı istediğimiz gibi okumaz programımız istediğimiz gibi çalışmaz.  <body ng>  <div>  <label>Name:</label>  <input type="text" ng-model="yourName" placeholder="Enter a name here"/>  <hr>  <h1>Hello **{{**yourName**}}**!</h1>  </div>  </body>  </html>  Ilk gün bunun gibi bir sürü alıştırma yaptım. Aşağıdaki program dizideki elemanları angular kodu ile nasıl tek seferde yazacağımızı gösteriyor.  <body data-ng-init="names=['ahmet','mehmet','veli']">  <form id="form1" runat="server">  <div>  <input type="text" data-ng-model="name" />**{{**name**}}**  <ul>  <li data-ng-repeat="personName in names">**{{**personName**}}**</li>  </ul>    </div>  </form>  </body> | |

|  |  |
| --- | --- |
| **Yapılan İş:** Angularım mimarisini anlama | **Tarih: 29.05.2015** |
| Angular MVC yapısına tamamen uymaktadır. Angular da Modul, wiev ve controller bulunmaktadır. Wiev iöinden scope ile bir very çekilebilir yada wievde conrollerdaki bir fonksiyon çalıştırılabilir.  Sadece Angular ile değil HTMl kısımına css yada javascriptin diğer kütüphaneleri ile de biçim verebiir dinamik uygulamalar geliştirebiliriz. Angular kodumuzu <Script> tagları arasına yazabildiğimiz gibi başka bir dosyaya da yazdırıp sayfada uzun kodlamaların önüne geçebiliriz ve daha sade bir kodlama elde edebiliriz. Performans olarak angular js çok iyidir ve singe page uygulamalarında çok başarılıdır.7  Örnek1:  <html xmlns="http://www.w3.org/1999/xhtml" data-ng-app="">  <head runat="server">  <script src="http://code.angularjs.org/snapshot/angular.js"></script>  <title></title>  </head>  <body data-ng-init="names=['ahmet','mehmet','veli']">  <form id="form1" runat="server">  <div>  <input type="text" data-ng-model="name" />**{{**name**}}**  <ul><li data-ng-repeat="personName in names">**{{**personName**}}**</li></ul>  </div>  </form>  </body>  </html>  Örnek 2:  <html xmlns="http://www.w3.org/1999/xhtml" data-ng-app="ad">  <head>  <title></title>  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.15/angular.min.js"></script>  <script type="text/javascript">  angular.module('ad', [])  .controller('abc', ['$scope', function ($scope) {  $scope.persons = [{ name: 'ismet', city: 'istanbul' },  { name: 'ahmet', city: 'karabük' },  { name: 'mehmet', city: 'ardahan' }];  }]);  </script>  </head>  <body>  <div data-ng-controller="abc">  <ul>  <li data-ng-repeat="per in persons">**{{**per.name**}}** : **{{**per.city**}}**</li>  </ul>  </div>  </body></html> | |

|  |  |
| --- | --- |
| **Yapılan İş:** Controller ile çalışma | **Tarih: 01.06.2015** |
| AngularJS uygulamaları temel olarak controller ve uygulama içindeki veriler ile çalışır. Bir controller ng-controller ile tanımlanır.  Örnek:  <div ng-app="" ng-controller="studentController">  ...  </div>  Bir JAvascript elemanı olan controller ile fonksiyonlar, özellikler tanımlanabilir. Tüm controller $scope bir parameter olarak Kabul ederler. Bu parameter ile controller ın kullanabileceği veriler html tarafından çekilebilir.  Örnek uygulama:  Angular kısmı :  <script>  function studentController($scope) {  $scope.student = {  firstName: "Mahesh",  lastName: "Parashar",  fullName: function() {  var studentObject;  studentObject = $scope.student;  return studentObject.firstName + " " + studentObject.lastName;  } }; }  </script>  Html kısmı:  Enter first name: <input type="text" ng-model="student.firstName"><br>  Enter last name: <input type="text" ng-model="student.lastName"><br>  <br>  You are entering: {{student.fullName()}}  Ekran çıktısı: | |
| **Yapılan İş:** Angularım mimarisini anlama | **Tarih: 02.06.2015** |
| Angular MVC yapısına tamamen uymaktadır. Angular da Modul, wiev ve controller bulunmaktadır. Wiev iöinden scope ile bir very çekilebilir yada wievde conrollerdaki bir fonksiyon çalıştırılabilir.  Sadece Angular ile değil HTMl kısımına css yada javascriptin diğer kütüphaneleri ile de biçim verebiir dinamik uygulamalar geliştirebiliriz. Angular kodumuzu <Script> tagları arasına yazabildiğimiz gibi başka bir dosyaya da yazdırıp sayfada uzun kodlamaların önüne geçebiliriz ve daha sade bir kodlama elde edebiliriz. Performans olarak angular js çok iyidir ve singe page uygulamalarında çok başarılıdır.7  Örnek1:  <html xmlns="http://www.w3.org/1999/xhtml" data-ng-app="">  <head runat="server">  <script src="http://code.angularjs.org/snapshot/angular.js"></script>  <title></title>  </head>  <body data-ng-init="names=['ahmet','mehmet','veli']">  <form id="form1" runat="server">  <div>  <input type="text" data-ng-model="name" />**{{**name**}}**  <ul><li data-ng-repeat="personName in names">**{{**personName**}}**</li></ul>  </div>  </form>  </body>  </html>  Örnek 2:  <html xmlns="http://www.w3.org/1999/xhtml" data-ng-app="ad">  <head>  <title></title>  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.15/angular.min.js"></script>  <script type="text/javascript">  angular.module('ad', [])  .controller('abc', ['$scope', function ($scope) {  $scope.persons = [{ name: 'ismet', city: 'istanbul' },  { name: 'ahmet', city: 'karabük' },  { name: 'mehmet', city: 'ardahan' }];  }]);  </script>  </head>  <body>  <div data-ng-controller="abc">  <ul>  <li data-ng-repeat="per in persons">**{{**per.name**}}** : **{{**per.city**}}**</li>  </ul>  </div>  </body></html> | |

|  |  |
| --- | --- |
| **Yapılan İş:** Ajax, htmldom öğrenme | **Tarih: 03.06.2015** |
| Angulardaki bazı directifler ile bir obje ye tıklandığında çeşitli fonksiyonlar çalıştırışabilir. Örneğin ;  ng-disable : bu direktif çalıştırıldığında buton yada obje ne ise onu inaktif eder ve arkasındaki fonksiyonun çaşışmasını engeller  ng-show: görünmez şekildeki yada inaktif durumdaki objeyi görünür yapar ve aktifleştirir.  ng-hide: görünür şekildeki bir objeyi görünmez yapar.  Ng-click: bir objeye tıklandığında arkada istenen bir fonksiyonu çalıştırır.  Örneğin:  <div ng-app="">  <table border="0"> <tr>  <td><input type="checkbox" ng-model="enableDisableButton">Disable Button</td>  <td><button ng-disabled="enableDisableButton">Click Me!</button></td>  </tr>  <tr>  <td><input type="checkbox" ng-model="showHide1">Show Button</td>  <td><button ng-show="showHide1">Click Me!</button></td>  </tr> Angular JS Tutorial  37  <tr>  <td><input type="checkbox" ng-model="showHide2">Hide Button</td>  <td><button ng-hide="showHide2">Click Me!</button></td>  </tr>  <tr>  <td><p>Total click: {{ clickCounter }}</p></td>  <td><button ng-click="clickCounter = clickCounter + 1">Click Me!</button></td>  </tr>  </table>  </div>  <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script> | |

|  |  |
| --- | --- |
| **Yapılan İş:** Ajax, htmldom öğrenme | **Tarih: 03.06.2015** |
| Angular JS $http servisi ile bir dosyadan verileri okuyabilir yanlız okunacak veriler JSON formatında olmalı, Örneğin: Controllerımız:  function studentController($scope,$http) {  var url="data.txt";  $http.get(url).success( function(response) {  $scope.students = response;  }); }  JSON dosyamız data.txt  [  {  "Name" : "Mahesh Parashar",  "RollNo" : 101,  "Percentage" : "80%"  },  {  "Name" : "Dinkar Kad",  "RollNo" : 201,  "Percentage" : "70%" Angular JS Tutorial  54  },  {  "Name" : "Robert",  "RollNo" : 191,  "Percentage" : "75%"  }]  Htmlimiz:  <table>  <tr>  <th>Name</th>  <th>Roll No</th>  <th>Percentage</th>  </tr>  <tr ng-repeat="student in students">  <td>{{ student.Name }}</td>  <td>{{ student.RollNo }}</td>  <td>{{ student.Percentage }}</td>  </tr>  </table> | |

|  |  |
| --- | --- |
| **Yapılan İş:** servis ve factory ile çalışma | **Tarih: 04.06.2015** |
| Angular js servis mimarisini desteklemektedir. Servisler Javascript içerisinde özel görevler için kodlanabilir. Angular js bir sürü inbuilt servisini barındırır. Örneğin $http, $route, $window, $location vb.  Bir servis iki şekilde oluştırılabilir, factory ve service medodu ile.  En genel örneğini aşağıda görebilirsiniz.  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <title></title>  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.15/angular.min.js"></script>  <script>  var app = angular.module("myapp", []);  app.controller("myController", function ($scope, calService) {  $scope.square = function () {  $scope.result = calService.square($scope.number);  }  });  app.service("calService", function(mathService){  this.square = function (a) {  return mathService.multiply(a,a);  }  });  app.factory("mathService" ,function(){  var factory = {};  factory.multiply = function (a, b) { return a \* b }  return factory;  });  </script>  </head>  <body ng-app="myapp">  <div ng-controller="myController">  <input type="number" ng-model="number" /><button ng-click="square()">X <sup>2</sup></button><br />  <h2>**{{**result**}}**</h2>  </div>  </body>  </html> | |

|  |  |
| --- | --- |
| **Yapılan İş:** view ile çalışma | **Tarih: 05.06.2015** |
| Angular js tek sayfa uygulamalarını desteklemektedir. Bu yüzden angular JS ng-view, ng-template directives ve $routeProvider servislerini desteklemektedir.  Örneğin aşağıdaki senaryoda add studenta tıklandığında addstudent controller ı ve adstudent templeti çalıştırılır. Add studenty ile yeni öğrenci eklenebilirken view student ile mevcut öğrenciler görüntülenebilir. Bu döngü tek sayfa içinde döndürülür.  <body>  <div ng-app="myapp">  <ul>  <li> <a href="#addStudent"> add student</a></li>  <li> <a href="#viewStudent">view student</a></li>  </ul>  <div ng-view></div>  <script type="text/ng-template" id="addStudent.htm">  <h2> view students</h2>  **{{**message**}}**  </script>  <script type="text/ng-template" id="viewStudent.htm">  <h2>view students</h2>  **{{**message**}}**  </script>  </div>  <script>  var app = angular.module("myapp", ['ngRoute']);  app.config(function ($routeProvider) {  $routeProvider.  when('/addStudent', {  templateUrl: 'addStudent.htm',  controller: 'AddStudentController'  }).  when('/viewStudent', {  templateUrl: 'viewStudent.htm',  controller: 'ViewStudentController'  }).  otherwise({  redirectTo: 'addStudent'  });  });  app.controller("AddStudentController", function ($scope) {  $scope.message = "add student form page";  });  app.controller("ViewStudentController", function ($scope) {  $scope.message=" here is wiew student page "});  </script>  </body>  </html> | |

|  |  |
| --- | --- |
| **Yapılan İş:** Servisler **;**$location, $anchorScroll | **Tarih: 08.06.2015** |
| Bu servisler ile bir sayfanın herhangi bir yeri işaretlenebilir ve sayfa işaretlenen yere kaydırılabilir.  <!DOCTYPE html>  <html ng-app="exampleApp">  <head>  <title></title>  <script src="http://code.angularjs.org/snapshot/angular.js"></script>  <script>  var app = angular.module("exampleApp", []);    app.controller("defaultCtrl", function ($scope, $location, $anchorScroll) {  $scope.show = function (id) {  $location.hash(id);  }  });  app.directive("compileDirective", function ($compile) {  return function (scope, element, attrs) {  var content = "<b>compile direktifi</b>"  var listElem = angular.element(content);  var compileFn = $compile(listElem);  compileFn(scope);  element.append(listElem);  }  function getData($timeout, $q) {  return function () {  var defer = $q.defer()  // simulated async function  $timeout(function () {  defer.resolve('data received!')  }, 2000)  return defer.promise  }  }  });  app.directive("interpolateDirective", function ($scope, $interpolate) {  return function (scope, element, attrs) {  var content = "<b>compile direktifi</b>"  var listElem = angular.element(content);  var compileFn = $interpolate(listElem);  compileFn(scope);  element.append(listElem);  }  });  app.directive("parseDirective", function ($parse) {  return function (scope, element, attrs) {  var content = "<b>compile direktifi</b>"  var listElem = angular.element(content);  var compileFn = $parse(listElem);  compileFn(scope);  element.append(listElem);  }  });    </script>  </head>  <body ng-controller="defaultCtrl">  <div style="position:fixed; margin-left:70%">  <button ng-click="show('bottom')">Go to Bottom </button><br /><br />  <button ng-click="show('middle')">Go to Middle </button><br /><br />  <button ng-click="show('top')">Go to Top</button><br /><br />  <span><a href="index.html">come back to index</a></span><br /><br />    </div>  <div style="margin:auto">  <p id="top"></p>  <ul><li>top</li></ul>  <span compile-directive></span>    </div>  <div style="margin-top:30%">  <p id="middle"></p>  <ul><li>middle</li></ul>  <span interpolate-directive></span>    </div>    <div style="margin-top:60%">  <ul><li>  <p id="bottom"></p>bottom</li></ul>  <span parse-directive></span>  </div>  </body>  </html> | |

|  |  |
| --- | --- |
| **Yapılan İş:** Servisler;$parse, $interpolate, $compile | **Tarih: 09.06.2015** |
| Angular JS in güzel taraflarından biride biride binlerce satır html kodunu bu servisler ile üretmemizi ve bazen gerekli olmayan html kodlarını bir controller ve servis yardımıyla kolayca eklememizi veya çıkarmamızı veya generate etmemizi sağlar.  <html xmlns="http://www.w3.org/1999/xhtml" ng-app="myapp">  <head>  <title></title>  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <script src="https://code.angularjs.org/1.3.8/angular.min.js"></script>  <script>  var app = angular.module('myapp', []);  app.controller('MainCtrl', function ($scope, $parse, $interpolate, $compile) {  $scope.name = 'Turkey';  //$parse  $scope.parse = $parse('name')($scope);  $scope.button\_text = 'click me';  //interpolate  // var html = ("<button ng-click='doSomething()'>{{label}}</button>");  // $scope.interpolate = $interpolate(html)($scope);    //$compile  $scope.compile = $compile(html)($scope);  return {  link: function (scope, element) {  $interpolate(element.html('<button ng-click="myclick()"> click me </button>')); }  }  }    );  app.directive("direkTif", function () {  return {  link: function (scope, element) {  element.html('<button ng-click="myclick()"> click me </button>');  } }  });  </script>  </head>  <body ng-app='MainApp'>  <div ng-controller='MainCtrl'>  <div>  <div>Parse: **{{**parse**}}**</div>  <!-- <div>Interpolate: {{interpolate}}</div>-->  <div>Compile : **{{**$compile**}}**</div>  <div direk-tif></div>    </div>  </div>  <ul style="margin-left:50%">  <li><a href="index.html">come back to index</a></li>  </ul>  </body>  </html> | |
| **Yapılan İş:** Kronometre uygulaması yapma | **Tarih: 10.06.2015** |
| Egzersiz olarak bir kronometre yapamam istendi. Bunun için $interval, $on servislerine ihtiyacımız var. uygulamanın Angular kodu aşağıdaki gibidir.  app.controller('StopwatchController', ['$scope', '$interval',  function ($scope, $interval) {  $scope.format = 'M/d/yy h:mm:ss a';  $scope.blood\_1 = 100;  $scope.blood\_2 = 120;  \*\* burada saatimiizn hangi formatta olduğunu belirliyoruz  var stop;  $scope.fight = function () {  // Don't start a new fight if we are already fighting  if (angular.isDefined(stop)) return;  stop = $interval(function () {  if ($scope.blood\_1 > 0 && $scope.blood\_2 > 0) {  $scope.blood\_1 = $scope.blood\_1 - 3;  $scope.blood\_2 = $scope.blood\_2 - 4;  } else {  $scope.stopFight();  }  }, 100);  };  $scope.stopFight = function () {  if (angular.isDefined(stop)) {  $interval.cancel(stop);  stop = undefined; } };  $scope.resetFight = function () {  $scope.blood\_1 = 100;  $scope.blood\_2 = 120; };  $scope.$on('$destroy', function () {  // Make sure that the interval is destroyed too  $scope.stopFight();  }); }])  .directive('myCurrentTime', ['$interval', 'dateFilter',  function ($interval, dateFilter) {  // return the directive link function. (compile function not needed)  return function (scope, element, attrs) {  var format, // date format  stopTime; // so that we can cancel the time updates  // used to update the UI  function updateTime() {  element.text(dateFilter(new Date(), format));  }  scope.$watch(attrs.myCurrentTime, function (value) {  format = value;  updateTime();  });  stopTime = $interval(updateTime, 1000);  element.on('$destroy', function () {  $interval.cancel(stopTime);  });  }  }]); | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik bir uygulama geliştirme | **Tarih: 11.06.2015** |
| Stajımın ana hedefi olan dinamik takvimi yapmam için öğreneme gerekn konulardan biride dynamic olaylar yapmak, farklı servisler ile aynı anda çalışmak.  Yaptığım aşağıdaki uygulama ile sayfanın ortasından başına, yada herhangi bir yerinden başka bir yerine sayfa kaydırılabilir ve bu aralarda da başka servvisler yardımıyla uygulamalar geliştirilebilir.  <!DOCTYPE html>  <html ng-app="exampleApp">  <head>  <title></title>  <script src="http://code.angularjs.org/snapshot/angular.js"></script>  <script>  var app = angular.module("exampleApp", []);    app.controller("defaultCtrl", function ($scope, $location, $anchorScroll) {  $scope.show = function (id) {  $location.hash(id);  }  });  app.directive("compileDirective", function ($compile) {  return function (scope, element, attrs) {  var content = "<b>compile direktifi</b>"  var listElem = angular.element(content);  var compileFn = $compile(listElem);  compileFn(scope);  element.append(listElem);  }  function getData($timeout, $q) {  return function () {  var defer = $q.defer()  // simulated async function  $timeout(function () {  defer.resolve('data received!')  }, 2000)  return defer.promise  }  }  });  app.directive("interpolateDirective", function ($scope, $interpolate) {  return function (scope, element, attrs) {  var content = "<b>compile direktifi</b>"  var listElem = angular.element(content);  var compileFn = $interpolate(listElem);  compileFn(scope);  element.append(listElem);  }  });  app.directive("parseDirective", function ($parse) {  return function (scope, element, attrs) {  var content = "<b>compile direktifi</b>"  var listElem = angular.element(content);  var compileFn = $parse(listElem);  compileFn(scope);  element.append(listElem);  }  });    </script>  </head>  <body ng-controller="defaultCtrl">  <div style="position:fixed; margin-left:70%">  <button ng-click="show('bottom')">Go to Bottom </button><br /><br />  <button ng-click="show('middle')">Go to Middle </button><br /><br />  <button ng-click="show('top')">Go to Top</button><br /><br />  <span><a href="index.html">come back to index</a></span><br /><br />    </div>  <div style="margin:auto">  <p id="top"></p>  <ul><li>top</li></ul>  <span compile-directive></span>    </div>  <div style="margin-top:30%">  <p id="middle"></p>  <ul><li>middle</li></ul>  <span interpolate-directive></span>    </div>    <div style="margin-top:60%">  <ul><li>  <p id="bottom"></p>bottom</li></ul>  <span parse-directive></span>  </div>  </body>  </html> | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik takvime giriş, ilk deneme | **Tarih: 12.06.2015** |
| Stajımın ana hedefi olan dinamik takvime artık başladım. Ilk olarak gereksinimim olan ana kodları kodlama başladım  Dinamik takvimdeki en önemliözelliklerden birisi kullanıcının herhangi bir tarih aralığını seçme özgürlüğünün olmasıydı. Ancak buradaki en büyük sorunlardan biriside başta benim için alınan tarihi formatındaki veriyi nasıl kolay bir şekilde gün ay yıla bölüp sonar bunları saat cinsinden yazmaktı. Bunnun için Moment js kütüphanesi işimşzi kolayca görür.  Ilk denememi gene javascript kütüphanelerinden biri olan jquery ile denedim bu jquerry kodu ile mouse dinamik takvimden çoklu seçim yapabiliyor  Öncelikle angular kodumum bir kısmı aşağıdaki gibi şekillenmeye başladı.  var app = angular.module("myapp", []);  app.controller("maincontroller",function ($scope) {    });  app.directive("hourDay", function ($parse,$log) {  return function (scope, element, attrs) {  var saat = 23;  while (saat >=0) {  if(saat<10) var content = "<b> 0" + saat + " : 00</b><br>"  else var content = "<b>" + saat + " : 00</b><br>"  var listElem = angular.element(content);  var compileFn = $parse(listElem);  compileFn(scope);  element.append(listElem);  saat = saat - 1;  console.log("burası tamam");    }  }    });  app.directive("dayMounth", function ($parse) {  var mounthdays = [  { "name": "jan", "day": 31 },  { "name": "feb", "day": 28 },  ….  .  .  .  .  .  Jquerry kodumuz da aşağıdaki gibi olmaya başladı  window.onload = function () {  // Click coordinates  var x1, x2, y1, y2;  //Variable indicates wether a mousedown event within your selection happend or not  var selection = false;  // Global mouse button variables  var gMOUSEUP = false;  var gMOUSEDOWN = false;  sPos = $("#selection").position();  // Global Events if left mousebutton is pressed or nor (usability fix)  $(document).mouseup(function () {  gMOUSEUP = true;  gMOUSEDOWN = false;  });  $(document).mousedown(function () {  gMOUSEUP = false;  gMOUSEDOWN = true;  });  // Selection frame (playground :D)  $("#YDR-Frame").mousedown(function (e) {  selection = true;  // store mouseX and mouseY  x1 = e.pageX;  y1 = e.pageY;  });  // If selection is true (mousedown on selection frame) the mousemove  // event will draw the selection div  $('#YDR-Frame').mousemove(function (e) {  if (selection) {  // Store current mouseposition  x2 = e.pageX;  y2 = e.pageY;  // Prevent the selection div to get outside of your frame  //(x2+this.offsetleft < 0) ? selection = false : ($(this).width()+this.offsetleft < x2) ?  });  function matchPos(xmiddle, ymiddle) {  // If selection is done bottom up -> switch value  if (x1 > x2) {  myX1 = x2;  myX2 = x1;  } else {  myX1 = x1;  myX2 = x2;  }  if (y1 > y2) {  myY1 = y2;  myY2 = y1;  } else {  myY1 = y1;  myY2 = y2;  }  .  .  .  .  .  .  .  .  .  .  . | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik takvim: deneme2 | **Tarih: 13 .06.2015** |
| Ilk denemede istediğimiz performansı elde edememeiştik, ayrıca angular kodıu ile jquerry kodumuz birbiri ile tam uyum içşnde çalıştırılamamıştı.  Sonraki denememde sadece angular kodu ile çalışmak istedim amaprogramta multiselect komutu tam olarak çalışmadı  // Code goes here  var myModule = angular.module('myApp', []);  myModule.directive('uiSelectable', function ($parse) {  return {  link: function (scope, element, attrs, ctrl) {  scope.$on('clearselection', function (event, document) {  element.find('.ui-selected').removeClass('ui-selected')  });  element.selectable({  stop: function (evt, ui) {  var collection = scope.$eval(attrs.docArray)  var selected = element.find('div.parent.ui-selected').map(function () {  var idx = $(this).index();  return { document: collection[idx] }  }).get();  scope.selectedItems = selected;  scope.$apply()  }  });  } }  });  myModule.controller('TempController', ['$scope', '$rootScope', '$timeout', function ($scope, $rootScope, $timeout) {  $scope.documents = [1, 2, 3, 4, 5, 6];  $scope.selectedItems = [];  $scope.$watch('selectedItems', function (data) {  });  $scope.clearSelection = function () {  $scope.selectedItems = [];  $timeout(function () {  $rootScope.$broadcast('clearselection', '');  }, 100);  }  $scope.dosomething = function (index) {  console.log('You clicked ' + index);  }  }]); | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik takvim deneme 3 | **Tarih: 16.06.2015** |
| Daha önceki denemelerimde isytediğim performansı elde edememiştim bide bu sıtrada yakalandığım soğık algınlığıda odaklanmamı ciddi bir şekilde düşürdü.  Bu sefer anguların başka bir fonksiyonu ile denemeye karar verdim ,  Yazdığım kodun html kısmı aşağıadaki gibi oldu.  Daha önceki denemelerimde table mantığı ile saatleri yazdırmaya çalışmıştım anvak button mantığı ile yapmaya karar verdim. Çünkü butonlara parameter atamak tablelerden daka kolay olmakta buttonun kendi fonksiyonları ile bunu halledebiliyorsunuz.  <script src="http://code.angularjs.org/1.2.11/angular.js"></script>  <script src="jquerycode.js"></script>  <script src="angularcode.js"></script>  <link rel="stylesheet" href="angcss.css" />  </head>  <body class="container">  <div ng-controller="dayCalculateController">    < <form id="demoform" action="#" method="post">  <div eval-expression></div>  <br><button type="submit" class="btn btn-default btn-block">Submit data</button>  </form>  <script>  var demo1 = $('select[name="duallistbox\_demo1[]"]').bootstrapDualListbox();  $("#demoform").submit(function () {  alert($('[name="duallistbox\_demo1[]"]').val());  return false;  });  </script>    Bu sefer angular kodunu eski hali ile bıraktım  Jquerry kodunun bir kısmı ise aşağıdaki gibi oldu  ;(function ($, window, document, undefined) {  var pluginName = 'bootstrapDualListbox',  defaults = {  bootstrap2Compatible: false,  moveSelectedLabel: 'Move selected',  moveAllLabel: 'Move all',  removeSelectedLabel: 'Remove selected',  removeAllLabel: 'Remove all',  moveOnSelect: true,  preserveSelectionOnMove: false,  selectedListLabel: false,  nonSelectedListLabel: false,  helperSelectNamePostfix: '\_helper',  // selectorMinimalHeight: 600,  },    isBuggyAndroid = /android/i.test(navigator.userAgent.toLowerCase());  \*\*\*\*\*\*\*\* | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik takvim deneme 5 | **Tarih: 17.06.2015** |
| Daha önceki denememde button mantığını kullanmıştım ancak buda istediğim performansı vermedi bunun yanında buttonlara style atamakta zor olduğu için formattada çeşitli sıkıntılar yaşadım.  Bu seferki denememde options mantığını kullanmaya karar verdim optionslara parameter atamak çok kolay oluyor ve browserlarda da performansı oldukça yüksek bir html objesi.  Options istediğim performansı sağladı.  var myApp = angular.module('myApp', []);  //myApp.directive('myDirective', function() {});  //myApp.factory('myService', function() {});  myApp.directive('multiselectDropdown', [function () {  return function (scope, element, attributes) {  element = $(element[0]); // Get the element as a jQuery element  // Below setup the dropdown:  element.multiselect({  buttonClass: 'btn btn-small',  buttonWidth: '200px',  buttonContainer: '<div class="btn-group"/>',  maxHeight: 800,  enableFiltering: true,  enableCaseInsensitiveFiltering: true,  buttonText: function (options) {  if (options.length == 0) {  return element.data()['placeholder'] + ' <b class="caret"></b>';  } else if (options.length > 1) {  return \_.first(options).text  + ' + ' + (options.length - 1)  + ' more selected <b class="caret"></b>';  } else {  return \_.first(options).text  + ' <b class="caret"></b>';  }  },  onChange: function (optionElement, checked) {  optionElement.removeAttr('selected');  if (checked) {  optionElement.attr('selected', 'selected');  }  element.change(); }  });  // Watch for any changes to the length of our select element  scope.$watch(function () {  return element[0].length;  }, function () {  element.multiselect('rebuild');  });Devam ediyor kodumuz \*\*\*\* | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik takvim, multiselect optimizasyonu | **Tarih: 18.06.2015** |
| Dinamik takvimimizin en büyük performans sorunu yapan yönlerinden biriside istenilen aralığın keyfi olması, istemilen aralık bir yıldan fazla seçilince renelenecek alan çok fazla oluyor ve bu alan içinde seçim yapacak kişi herhangi bir tarihteki saati seçince multiselekt algoritmamız bu tarih aralıklarını seçif ramda tutması gerekiyor, buda kodun performansını oldukça etkiliyor.  Multiselect ve rendeleme için çeşitli alternative yollar düşünülebilir. Bunlardan bana en mantıklı gelen sanallaştırma algoritması. Bu algoritma ile sadece o an ekranda ki alan render edilir eğer seçilen alan daha fazla ise alttaki çubuk saga kaydıkça yeni alan rendelenir. Buda çok ciddi anlamda performansı artıran bir mantık.  Jquerry kodunun bir kısmı aşağıdaki gibi olmaktadır.  (function($) {  $.fn.overlap = function(args) {  args = $.extend({  rect: {x:0, y:0, w:0, h:0},  element: false,  mark: false,  unmark: true,  filterMark: function(index) {return true},  filterUnmark: function(index) {return true},  callback: function($elements) {},  w: function($element) {return $element.outerWidth(true)},  h: function($element) {return $element.outerHeight(true)},  position: function($element) {return $element.position()},  bring: true  }, args);  Object.prototype.overlap = function(rect) {  return(  (rect.x <= this.x + this.w && rect.x + rect.w >= this.x) &&  (rect.y <= this.y + this.h && rect.y + rect.h >= this.y)  );  };  if(args.element!=false) {  if(typeof(args.element)=="string") {  h: args.h($(this))  })  ) {  return true;  }  });  if(typeof(args.mark) == "string") {  if(args.unmark) {  this.filter(args.filterUnmark).removeClass(args.mark);  } else {  return(this);  .  .  .  . | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik takvim: mapping optimizasyonu | **Tarih: 19.06.2015** |
| Dinamik takvimizin en önemli özelliklerinden biriside seçtiğimiz aralıkları rapor şeklinde çıktı elde edilmesi ve bu çıktının istenildiği zaman tekrar geri okutulmasıdır. Ancak tarih aralığı keyfi olduğu için rapor oluşturulurken veya okunıurken referans alınacak tarih aralığı çok uzun olabilmekte bazı durumlarda milyonlarca elemanlı bir matrix olabilmektedir.  Bu durum performansı çok ciddi şekilde düşürmektedri. Bunun için yapılabilecek en önemli optimizasyonlardan bir taneside öapping optimizasyonları. Raporu rendelerken yada okurken tek satırlık bir arraydan maping yapmak yerine bu durum çok boyutlu arraylar ile yapılabilir. Bende algoritmamda gün ay yıl ve saatten oluşan multi dimensional bir mantık yürüttüm bu şekilde yıl içindeki 12 ay ve bunlar içindeki gün sayısına göre mapping olayını çok daha hızlı bir şekilde yapabilmekteydim.  while (mounth < 12) {  for (day = 1; day <= mounthlenghts[mounth].day; day++) {  var h = 24;  while (h>0) {  time = {  hour: h,  day: day,  "mounth": mounthlenghts[mounth].name,  }  arraya.push(time);  h--;    }  x++;      }  mounth++;  }  $scope.documents = arrayc.concat(arraya, arrayb);  console.log(x); | |

|  |  |
| --- | --- |
| **Yapılan İş:** Dinamik takvim: kodların düzenlenmesi ve eksiklikleri belirleme | **Tarih: 20.06.2015** |
| Stajımın son günü ana hedefim olan performansı ne kadar tutturduğumu uygulamanın eksik kalan yönlerini nasıl gidereceğimi ve ilerde kariyerime nasıl bir yön vermem gerektiğini staj sorumlum ile konuştum. Kodda fazla olan yada gereksiz performansı azaltan yerleriinceledim. Daha uygulamaya still vermediğim için görünüş olarak çok iyi değildi onu düzeltmeye çalıştım.  .  .  .  float: left;  }  section div.mark {  background: blue;  }  div.selection {  border: 1px solid royalblue;  background: rgba(65, 105, 225, 0.2);  position: absolute;  z-index: 9;  }  .  .  .  Optimize edilmeye çalışılmış multiselect kodlamalarının bir kısmı aşağıdaki gibi oldu..  $(window).load(function () {  var SECTION\_DIV = "section > div";  var $SELECTION = $("div.selection");  var $selected = $();  var point = [0, 0];  $(SECTION\_DIV).on("mousedown touchstart", function (e) {  if (e.which === 1 || e.type == "touchstart") {  if (!e.ctrlKey) {  $(SECTION\_DIV).removeClass("marked mark");  }  $(this).addClass("marked mark");  }  });  $("body").on("mousedown touchstart", function (i) {  if (i.type == "touchstart" || (i.which === 1 && !$(i.target).is(SECTION\_DIV))) {  if (!i.ctrlKey && !(i.type == "touchstart" && $(i.target).is(SECTION\_DIV))) {  $(SECTION\_DIV).removeClass("marked mark");  } else {  console.log(i.target);  }  if (i.type == "touchstart") {  i.pageX = i.originalEvent.touches[0].pageX;  i.pageY = i.originalEvent.touches[0].pageY;  }  if ((point[0] != i.pageX && point[1] != i.pageY) || i.type == "mousedown") {  point[0] = i.pageX;  point[1] = i.pageY;  $SELECTION.css({  "top": i.pageY,  "left": i.pageX  });  $SELECTION.removeClass("hide");  }  $(this).on("mousemove touchmove", function (f) {  if (f.type == "touchmove") {  f.pageX = f.originalEvent.touches[0].pageX;  f.pageY = f.originalEvent.touches[0].pageY;  }    .  .  . | |