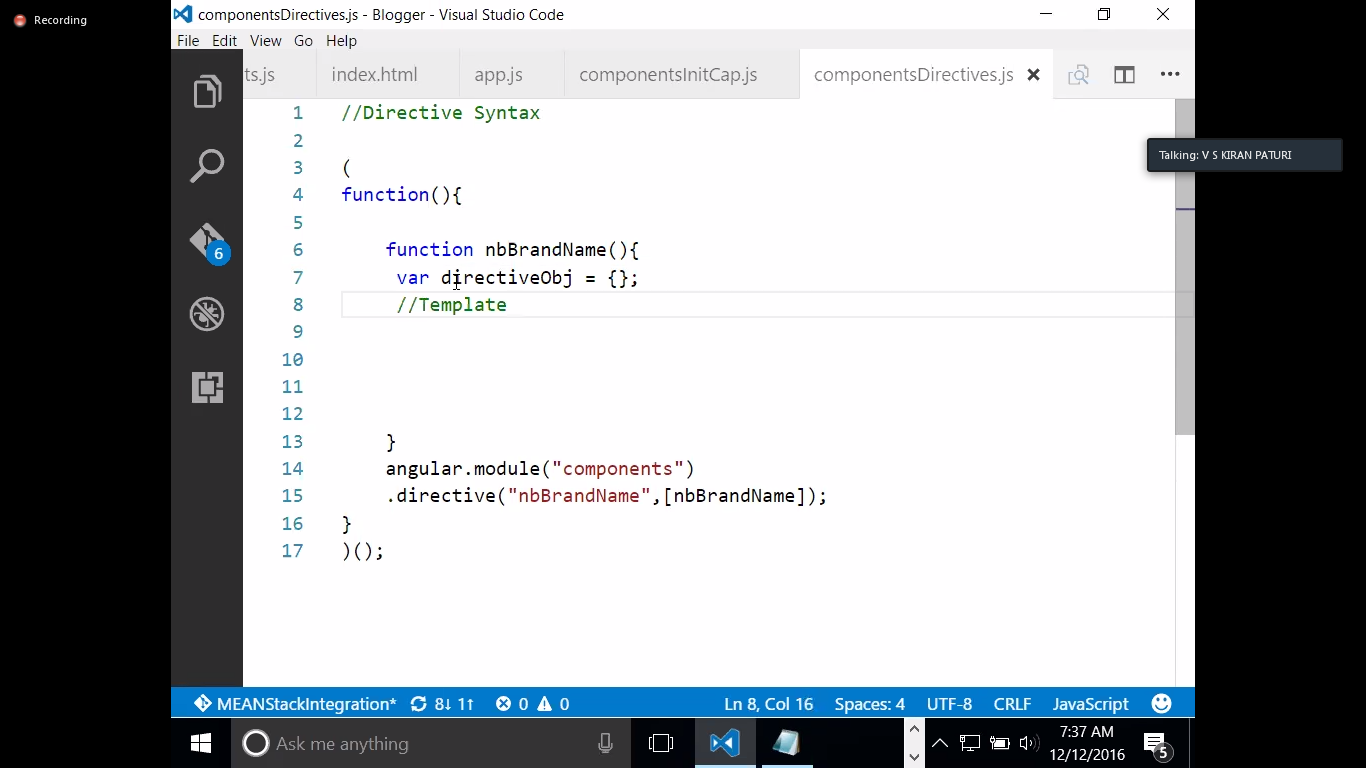
Directives

Directives are the reusable components in angular .They are the basic building blocks to make angular.

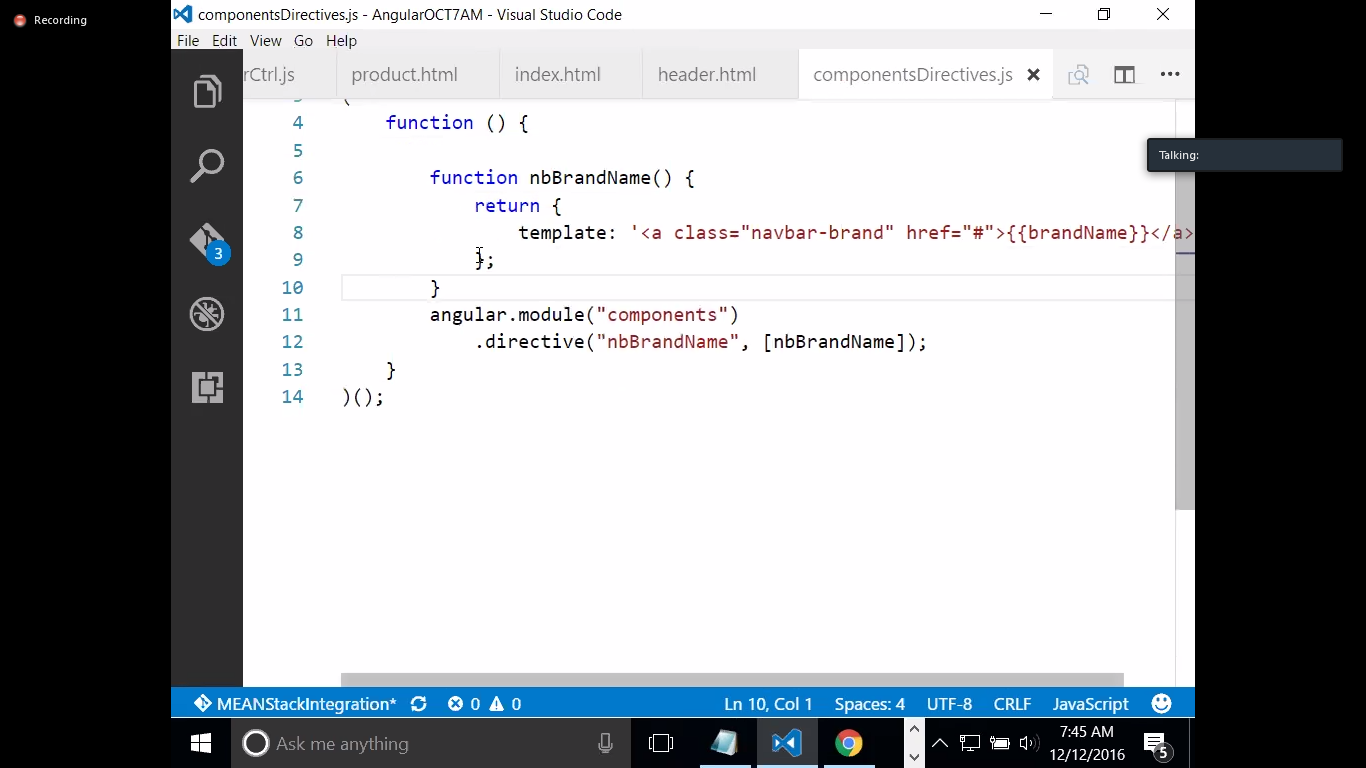
There are

* Functional Directives
* Validation Directives
* UI Rendering directives

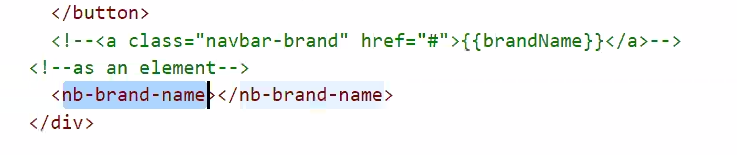
Directive:- Directives will return an object.



In a directive ,I you are referring Capital letters will converted to small letters with a hyphen(-)



In header.html



Element: Template,Template URL

Atribute: Restrict

Class:

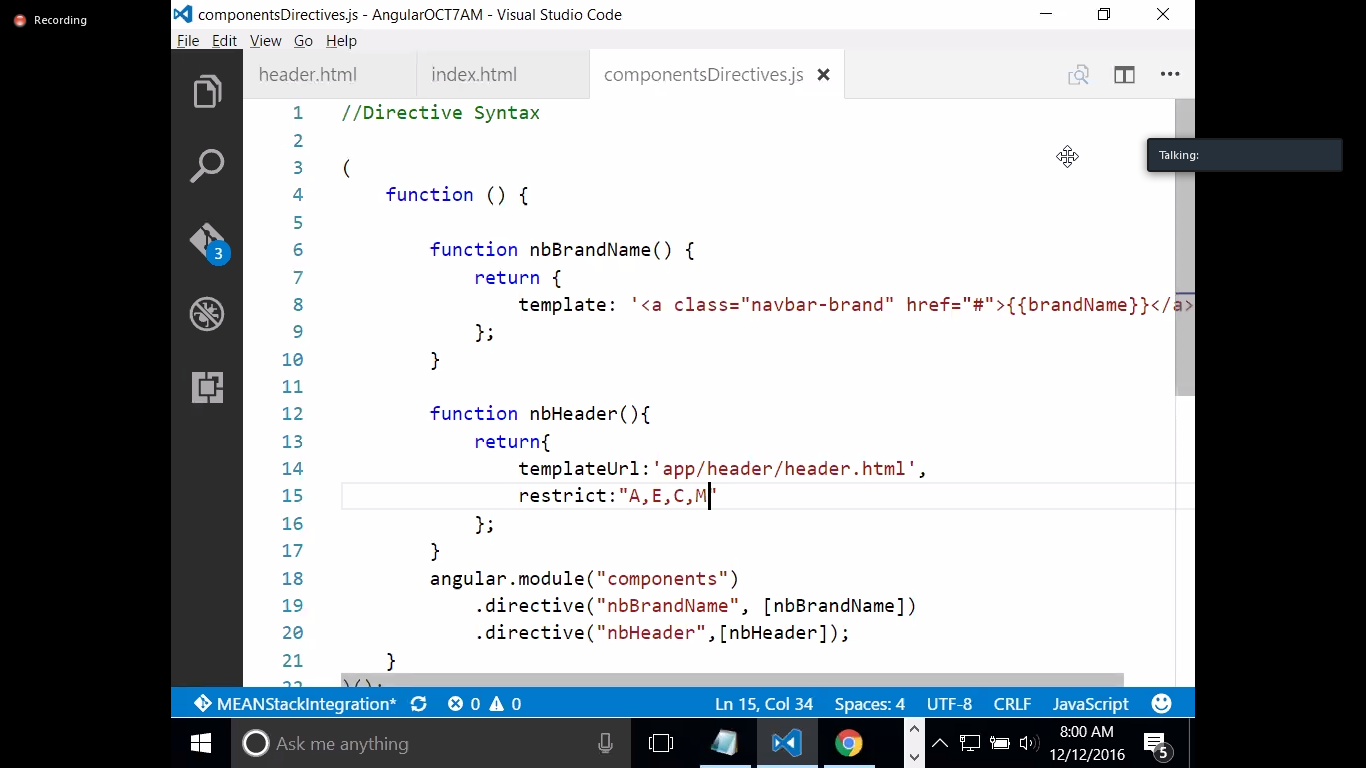
By Default,any directive can be used as an element and also as an attribute.

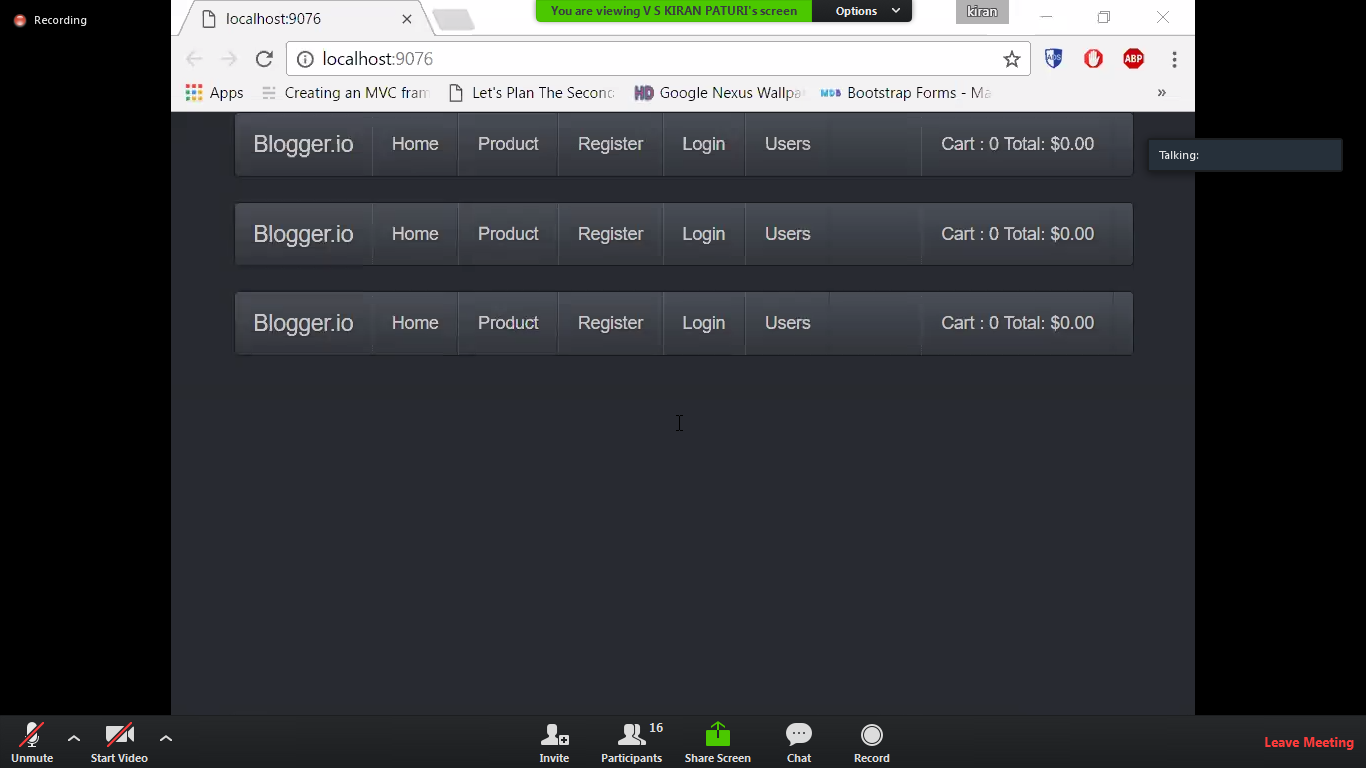
Restrict is another property of directive,which will help us to declare the directive as an element,attribute,class and comment

Elements are html tags

Attributes are properties of element, Preffered way is always to use attribute.

Directives will automatically inherit the controller



**Priorty :-**

Directive has priority prop, in case of multipl directivesin an elemnt,directv with highr prop will gets execute frst

**Functionality in our directive**

**Link Function:-**

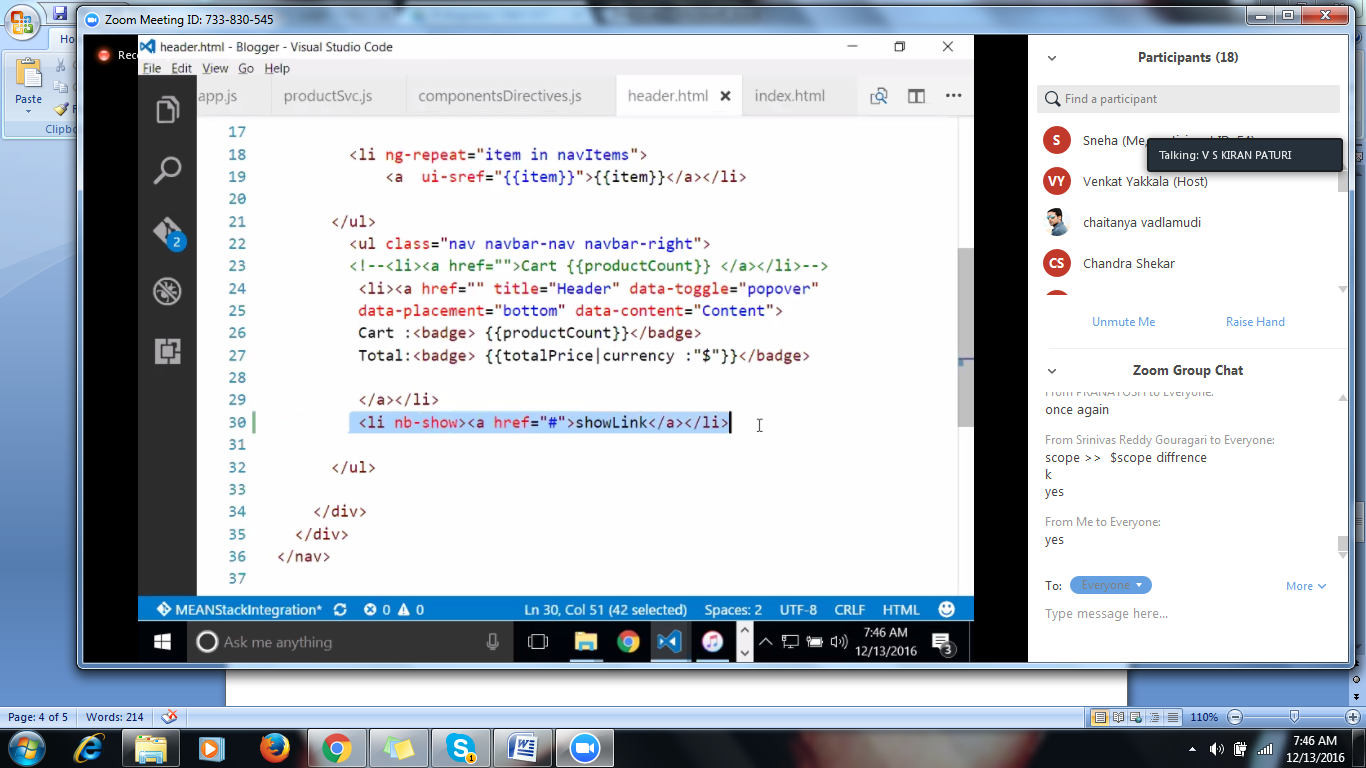
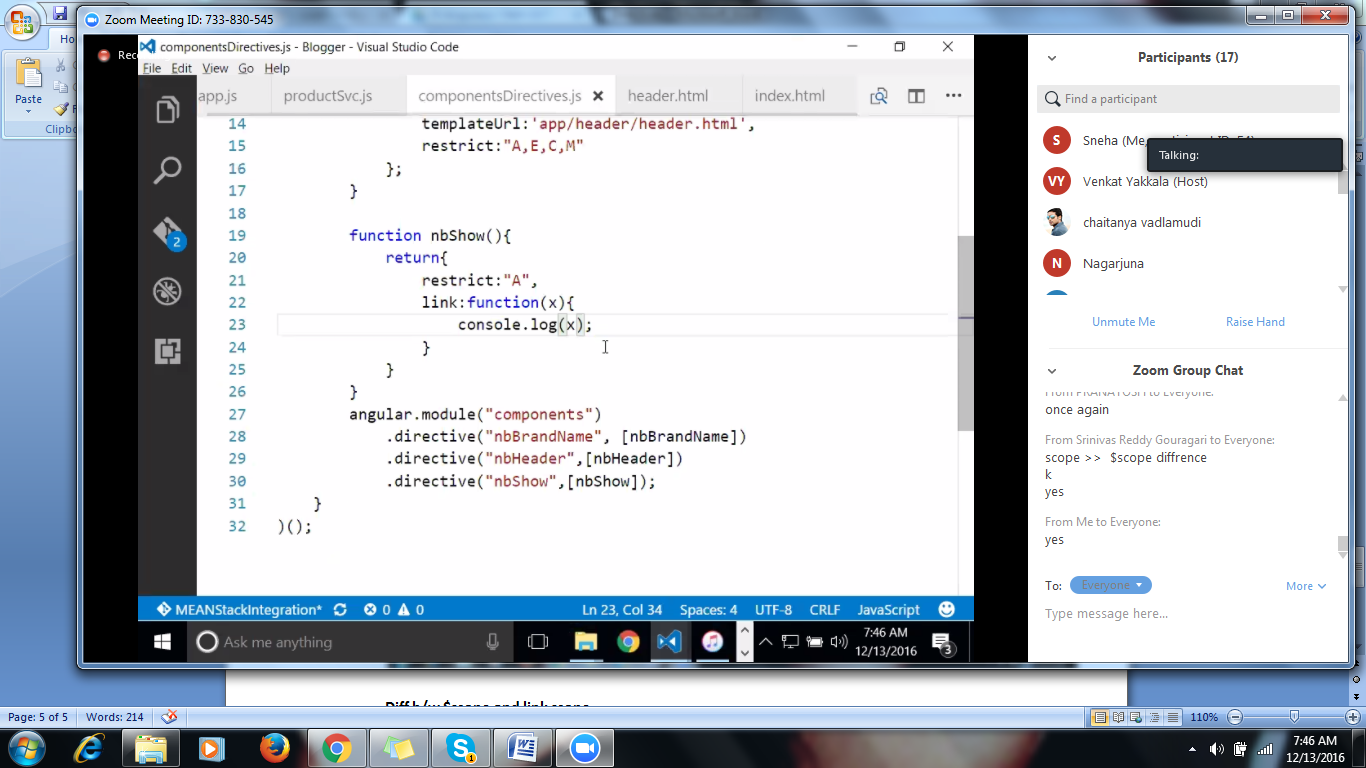
Link is another prop of a directive ,Typ of link is a function.Link Fn will take 3 params primarily.

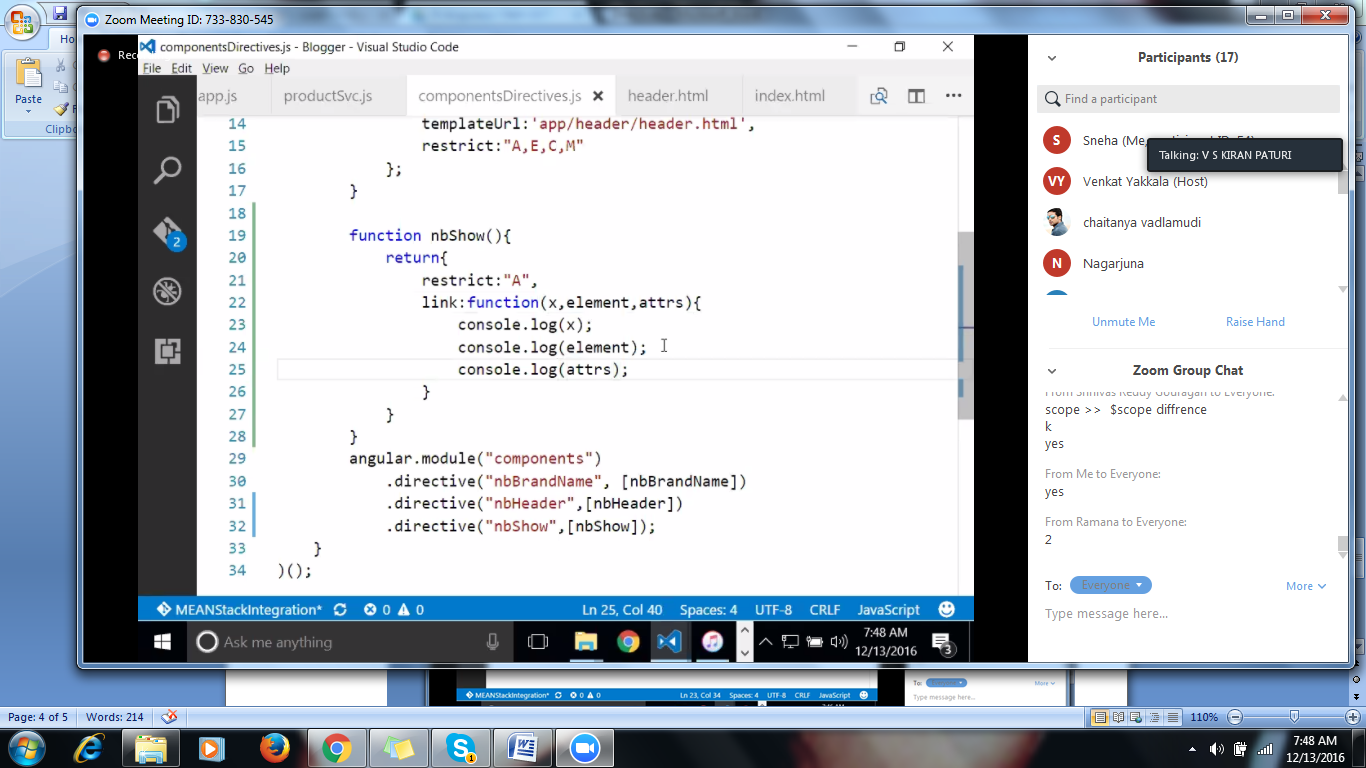
* 1.scope:- Scope is a parameter to the function ,usually this scope is inherited from the controller.
* 2.Element:-
* 3.Attribute

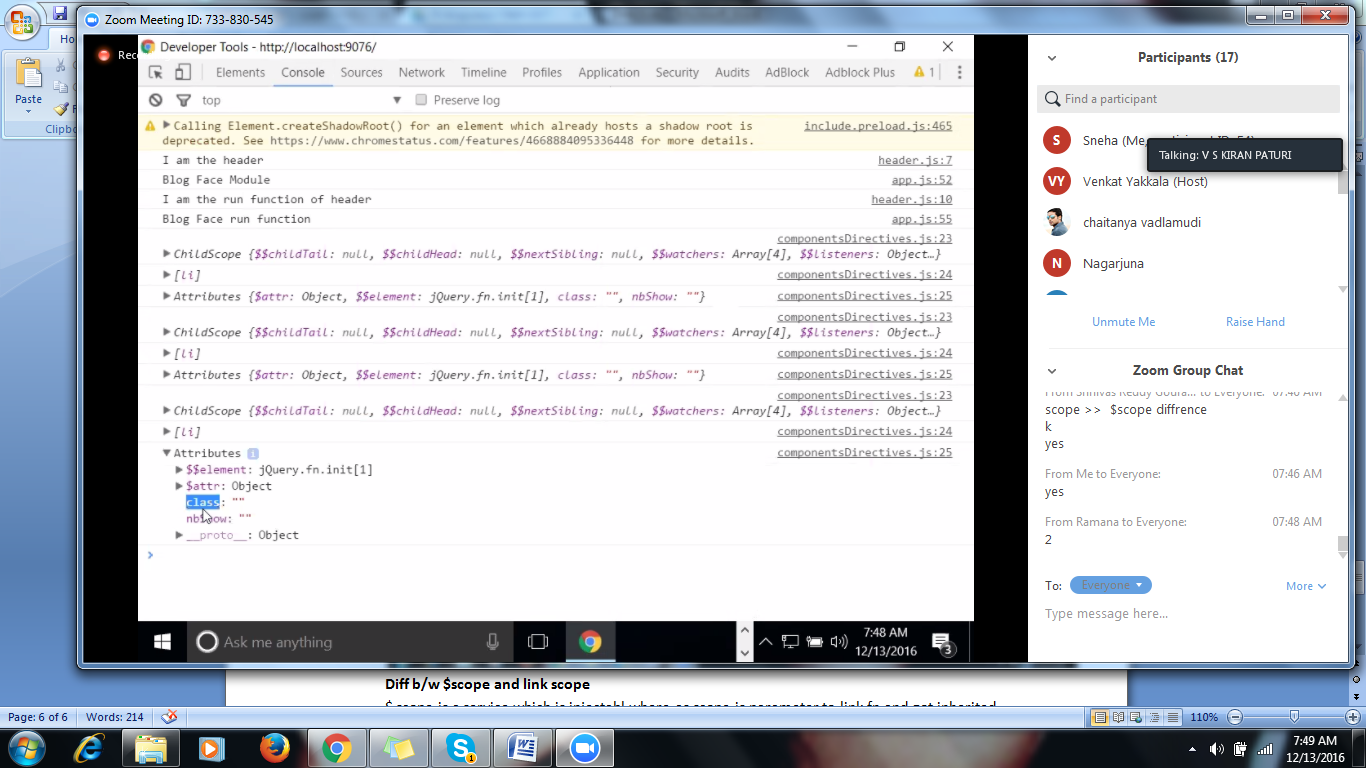
Link: function (scope ,Element ,Attribute){

}

This function will offer us functionality to directive.







**Diff b/w $scope and link scope**

$ scope is a service which is injectabl,where as scope is parameter to link fn and got inherited from the parents controller

**Plug-in to Directive conversion:-**

Converting existing Jquery plugin to Angular Directive:-

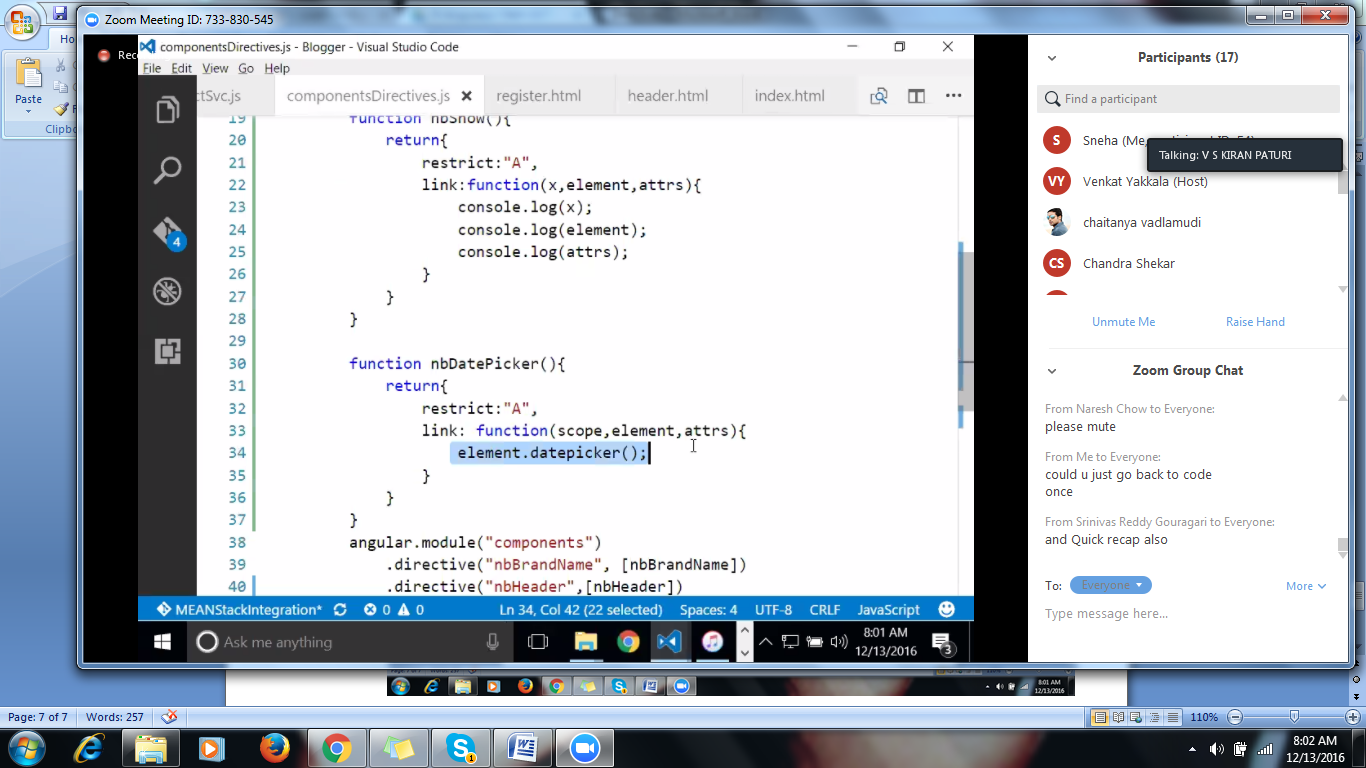
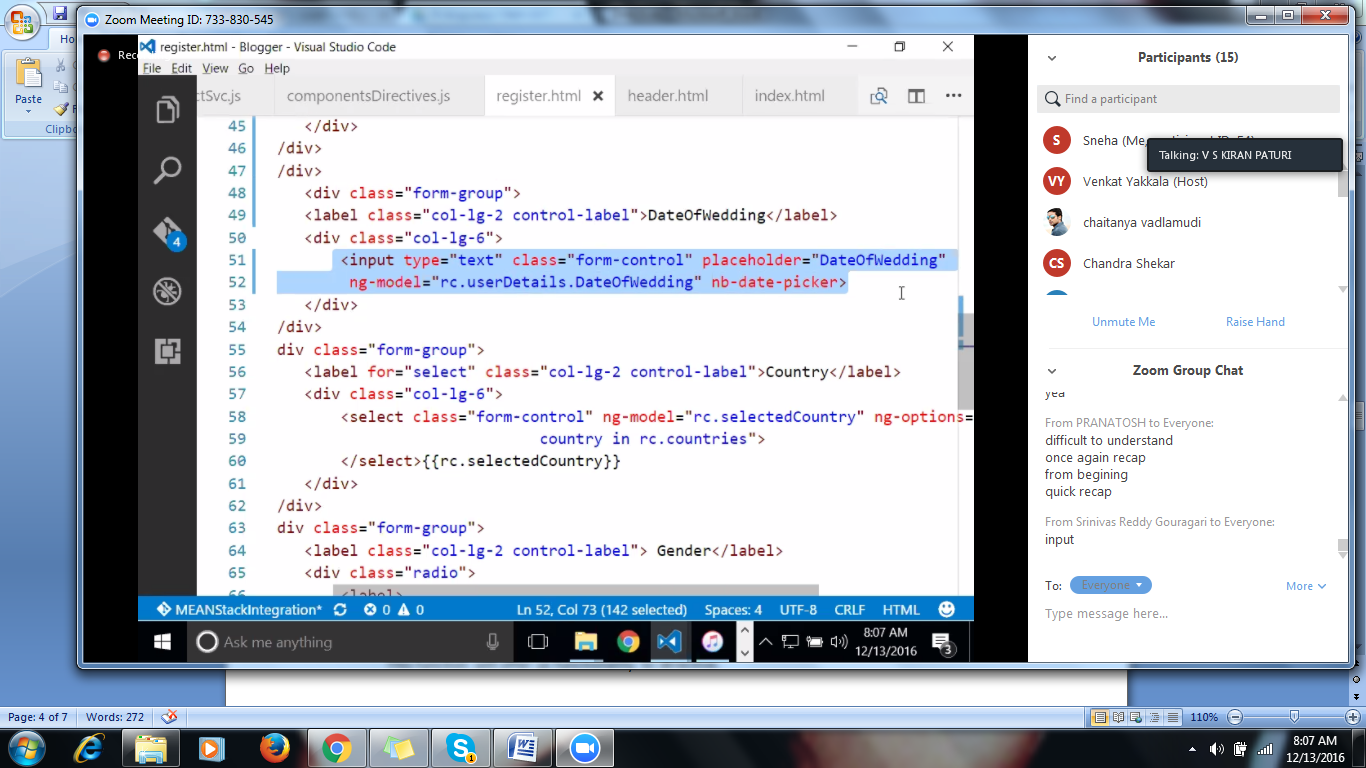
Ex:-Date Picker

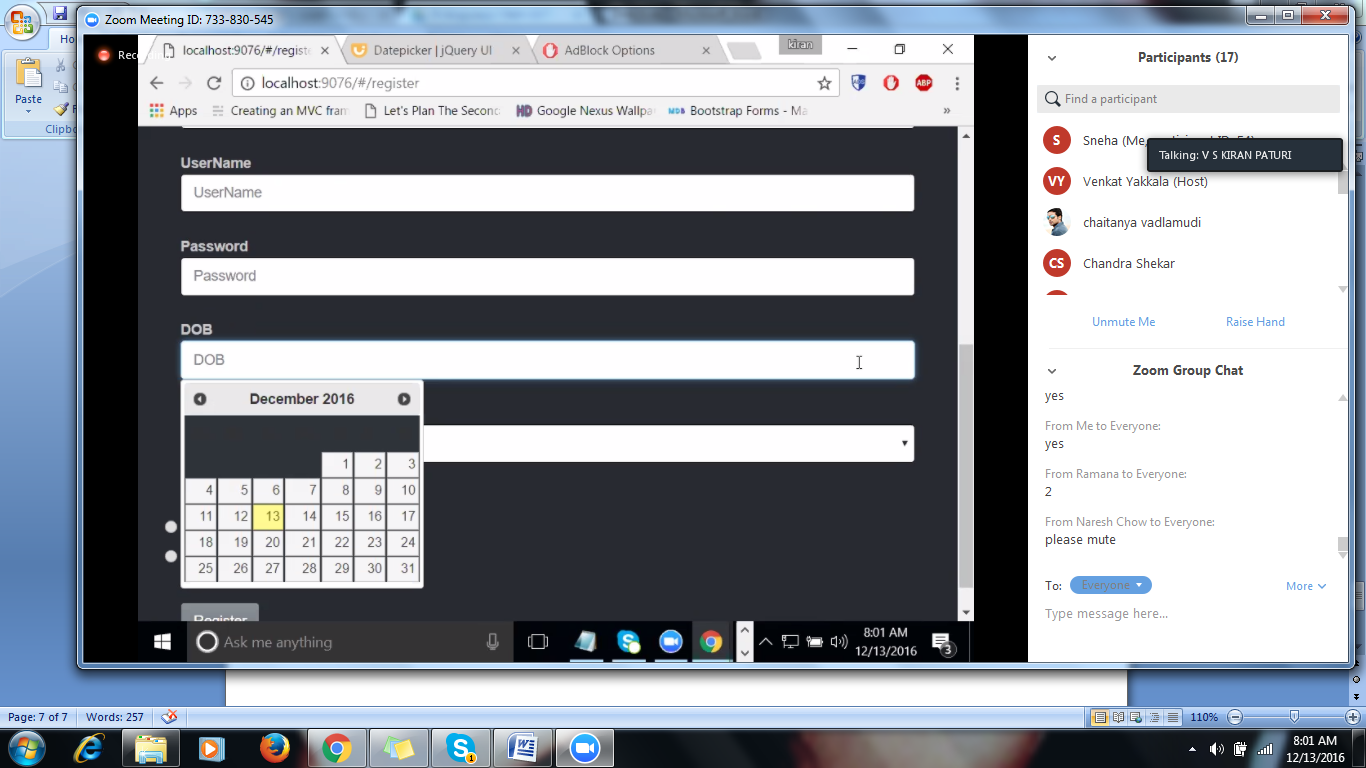
Step1:Try to understand plug-ins behavior

Step2:-Check the usage of the plug-in inside the code.

Step3:-Inside directive try to apply the function of plug-in to the element,.

Add the scripts and css files of jquery to our html

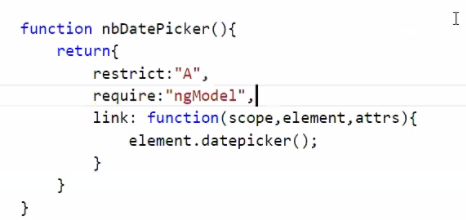
 



Require:-

Require is another property of a directive,that will make the mandatory requirements.

That mean if it is there,then only it will work,ex: if ngModel is there then only datepicker will work.



Priorty can be set as

Priorty:ngmodel;

nbEnter:-

$ eval is a function that will evaluate the scope properties.

Isolated scope:

By default directives inherit the controllers scope.Challng here is we r dependent on ctroller properties.In isolated scop we will use only directv scope.this scope is not inherited from controller.



Whenever we want to use re-usable components ,we will go for isolated scope.

but the problem is all the time we should know the properties,those directives are bound to the ctrller prop