# Class22: Quick Link Webpart

# Part:I

# How to make this kind of links

# 

# Look the ff code

   <div class="container">

        <div class="row">

<!--=================================

            <div class="col-sm-3">

                <div class="card">

                    <div class="card-body">

                        <a href="#">Link 1</a>

                    </div>

                </div>

            </div>

            <div class="col-sm-9"></div>

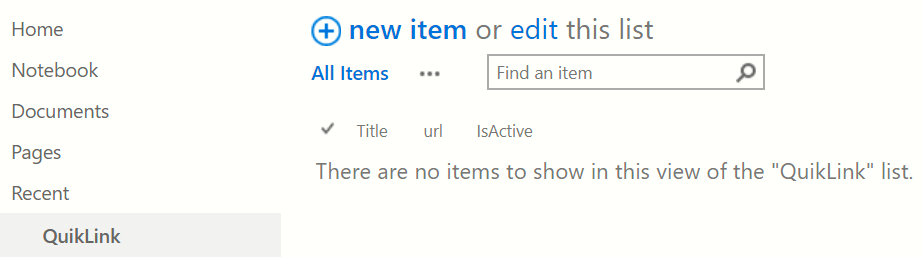
================================🡪

        </div>

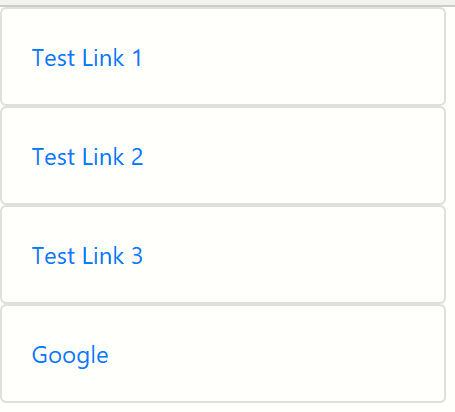
    </div>

# Reapt it as we want and then simply change the number.

# Create a List with the following coloumns



Simply by changing Link to Test Link we get the following.



**But this are hard code so we collapsed it and we**

 <div class="container-fuid">

        <div class="row">

            <div class="col-sm-3">

                <div class="card">

                    <div class="card-body">

                        <a href="#">Test Link 1</a>

                    </div>

                </div>

            </div>

            </div>

            </div>

**do <script></script>.and put the following code between them.**

 //Function to get list items by passing List name to the list

    function getItems(listName) {

        var url = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/getbytitle('" + listName + "')/items";

        return $.ajax({

            url: url,

            method: "GET",

            headers: { "Accept": "application/json; odata=verbose" }

        });

    }

Chanhe: getItems(listName) By getData(url)

**And delete**

 var url = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/getbytitle('" + listName + "')/items";

**then we get Like**

<script>

//Function to get list items by passing List name to the list

    function getDara(url) {

        return $.ajax({

            url: url,

            method: "GET",

            headers: { "Accept": "application/json; odata=verbose" }

        });

//all functions goes through

    }

</script>

**Like**

 <script>

         //Function to get list items by passing List name to the list

    function getData(url) {

        return $.ajax({

            url: url,

            method: "GET",

            headers: { "Accept": "application/json; odata=verbose" }

        });

    }

    $(document).ready(function(){

    });

    </script>

**Check the ff in YARD**

https://nmek.sharepoint.com/sites/Demo/\_api/web/lists/getbytitle('QuikLink')/

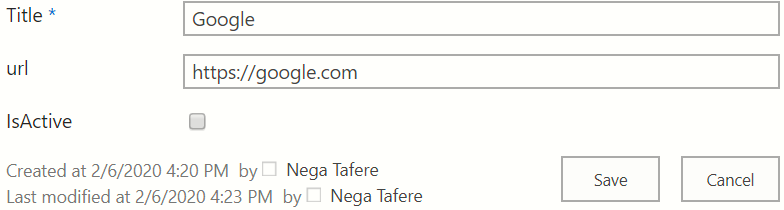
https://nmek.sharepoint.com/sites/Demo/\_api/web/lists/getbytitle('QuikLink')/items?

https://nmek.sharepoint.com/sites/Demo/\_api/web/lists/getbytitle('QuikLink')/items?$select = Title

1. $select = Title

2.$filter = IsActive eq true

3.$filter=IsActive ne false



So from the ff code we make some changes

  <script>

         //Function to get list items by passing List name to the list

    function getData(url) {

            return $.ajax({

            url: url,

            method: "GET",

            headers: { "Accept": "application/json; odata=verbose" }

        });

    }

    function getItems(listName){

    var url = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/getbytitle('" + listName + "')/items";

    }

    $(document).ready(function(){

    });

    </script>

The change is:

    var url = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/getbytitle('" + listName + "')/items";

after the ff step we can test on SP console

<script>

         //Function to get list items by passing List name to the list

    function getData(url) {

            return $.ajax({

            url: url,

            method: "GET",

            headers: { "Accept": "application/json; odata=verbose" }

        });

    }

    function getQuikLink(){

    var url = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/getbytitle('QuikLink')/items";

    return getData(url);

    }

    $(document).ready(function(){

        getQuikLink().then(function(response){

            console.log('Data is ',response);

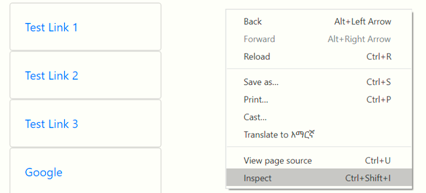
            var results=response.d.results;

            //end of implementation

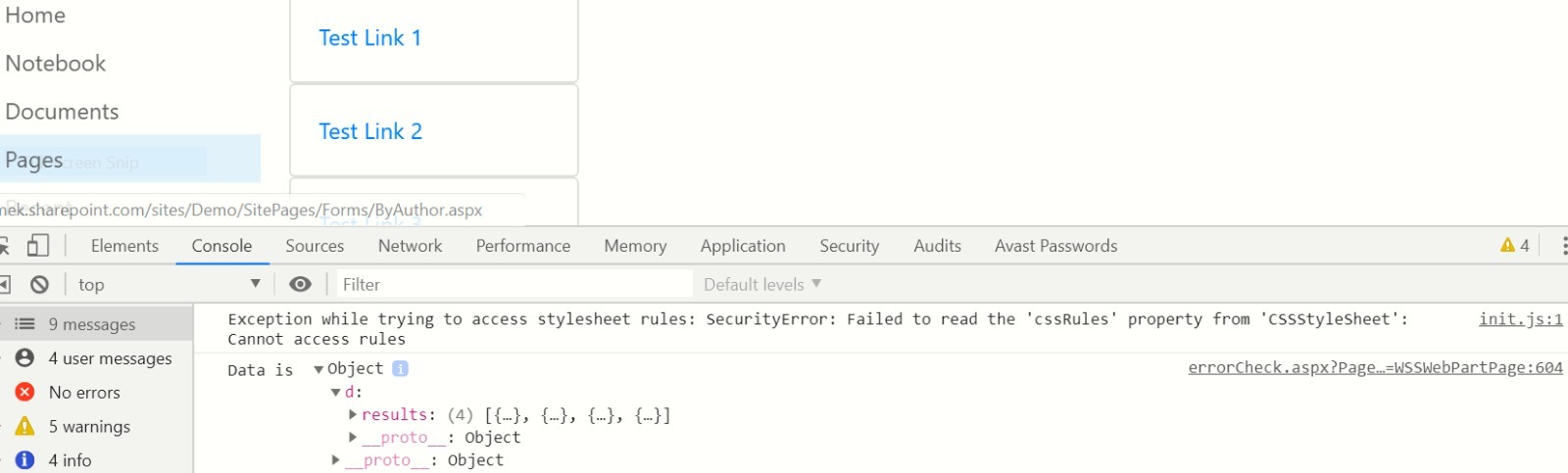
        },function(error){console.log('oops error',error)});

    });

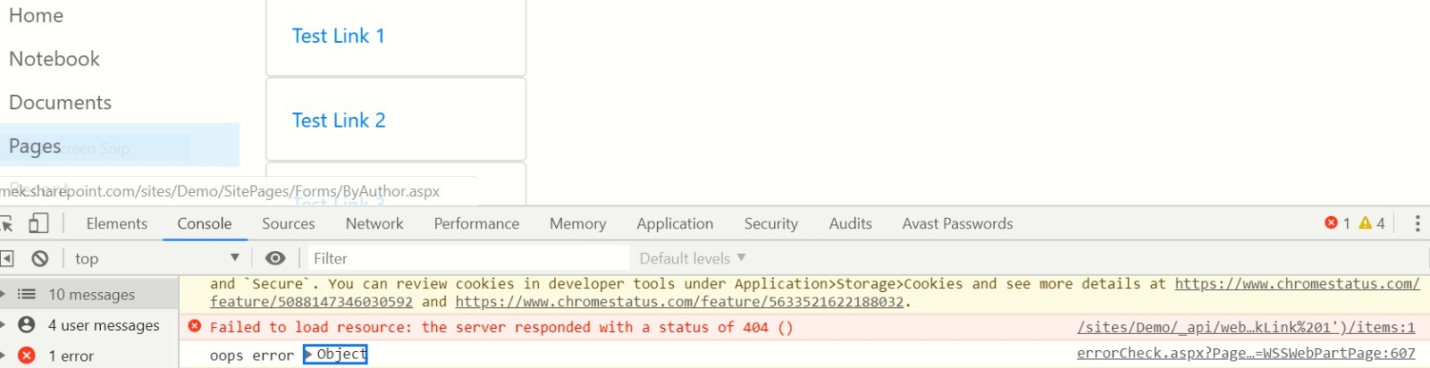
    </script>



**Then we get result**



Let us ask unexisting List eg “QuikLink 1” then test it then we get



**After this we token about how we do display.**

**By adding:**

 for(i in results){

                console.log(results[i].Title);

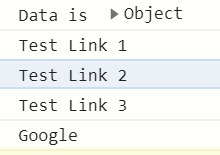
            }

**Or**

results.forEach(i){

                console.log(results[i].Title);

**and then by testing it we get**

****

**Then delete the card between row blse that is hard code and we put**

<div class="card mb-1">

    <div class="card-body">

        <a href="#">Link 1</a>

    </div>

</div> //we can change it to

<!--put results[i].Title instead of Link 1-->

'<div class="card mb-1"><div class="card-body">' +

'<a href="' + results[i].url +'">'+ results[i].Title +'</a></div></div>';

**By doing the above changes and put the change in the console we get the previous html.**

**That is we can log like by adding some changes from the previous code**

 var myHtml = '';

                for (i in results) {

                    console.log(results[i].Title); //Title colmn

                    myHtml += '<div class="card mb-1"><div class="card-body">' +

                        '<a href="' + results[i].url + '">' + results[i].Title + '</a></div></div>';

                }

                console.log(myHtml);

**(+) next to myHtml is very important**

**By logging this we generate the ff**

****

<div class="card mb-1">

    <div class="card-body"><a href="#">Test Link 1</a></div>

</div>

<div class="card mb-1">

    <div class="card-body"><a href="#">Test Link 2</a></div>

</div>

<div class="card mb-1">

    <div class="card-body"><a href="#">Test Link 3</a></div>

</div>

<div class="card mb-1">

    <div class="card-body"><a href="https://google.com">Google</a></div>

</div>

**Generated and by putting it in , NB we insert it by the ff equestion**

<div id="QuikLink">  </div>

**We get the previous html. After once log it we make //**

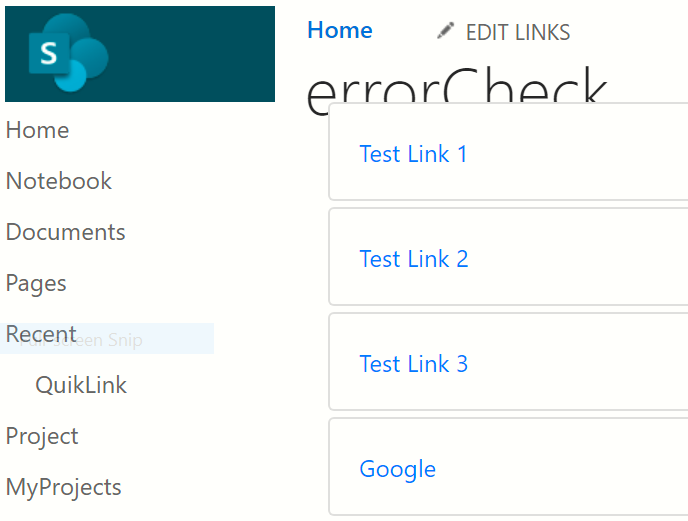
//console.log(myHtml);

**Instead we use the ff and log it**

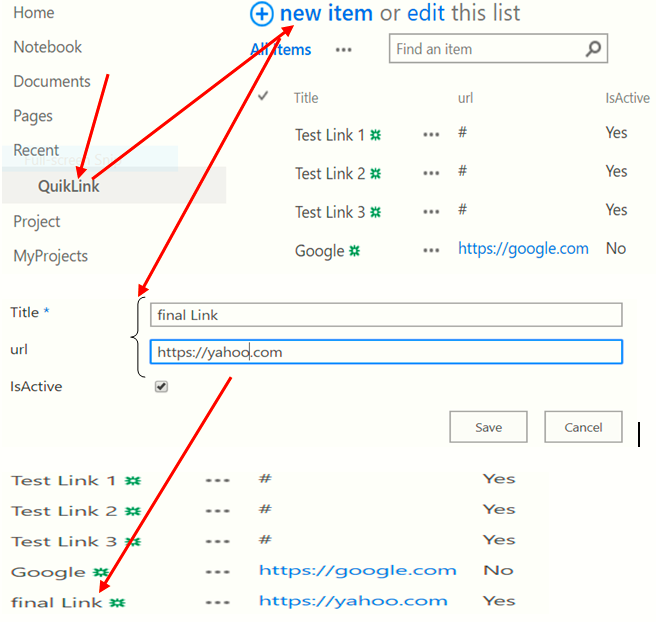
$('#QuikLink').html(myHtml);

**(#) is important.**

**Then we get the following result**

****

**If we want to add another item**

****

**#How to display the title of the List/project. Simply by changing the title that is**

**var url = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/getbytitle('QuikLink')/items";**

QuikLink

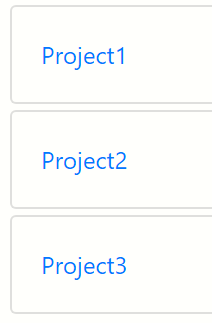
**Changed by**

myProjects

**and**

' + results[i].url + '

**Changed by # b/se we don’t have Url. Then log it . then we get**

****

console.log(results[i].Title);

**This console is important to get the Title.**

**You see: to get🡺Title use** results[i].Title

**to get🡺Budget use** results[i].Budget

**to get🡺status use** results[i].status

**Adding some CSS codes**

**Card body hover**

<style>

    .card-body.p-3:hover{

        background-color: darkgray;

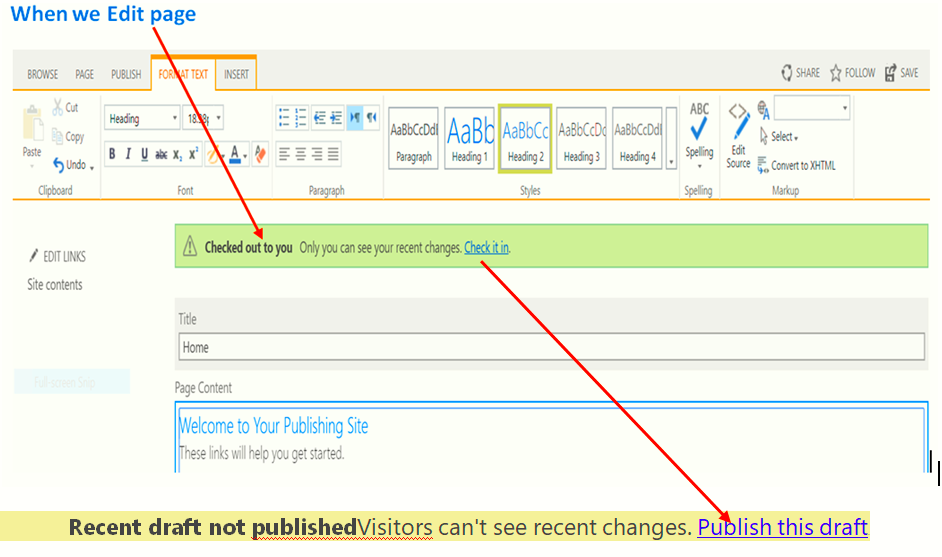
    }

</style>

**Part:II**

**Create publish site:To use master page,Design manager,…..check them under site setting**

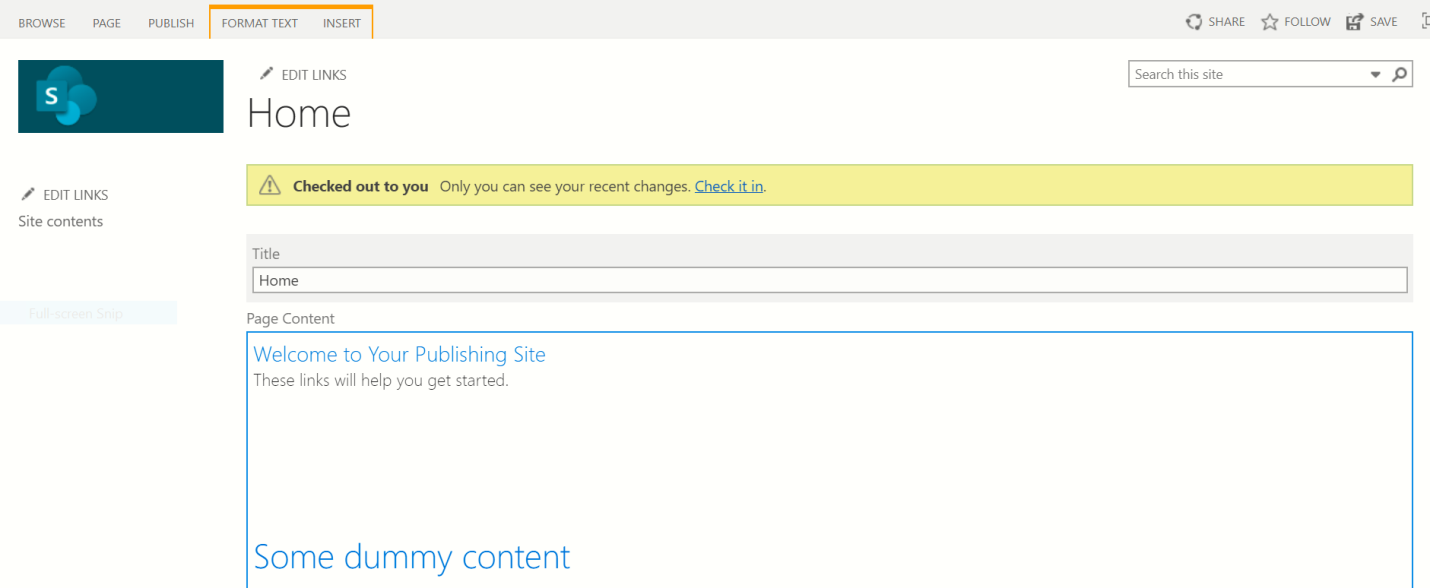
**When we Edit page**

****

**Design our master page:**

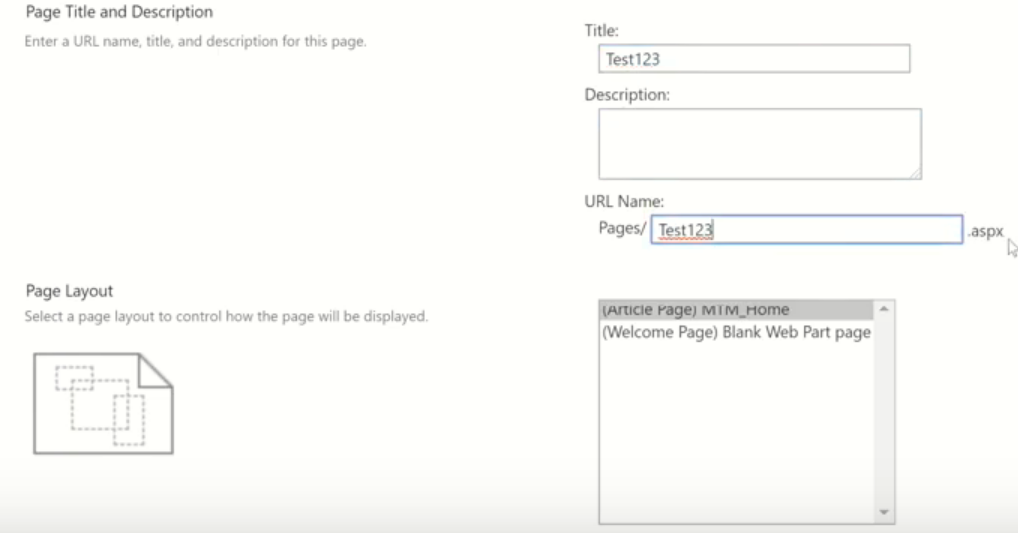
**Publishing page: made from Master page, page layout and and from page content.**

**When we edit this page we cann’t edit the top,left navigation b/se those are created by master page.**

****

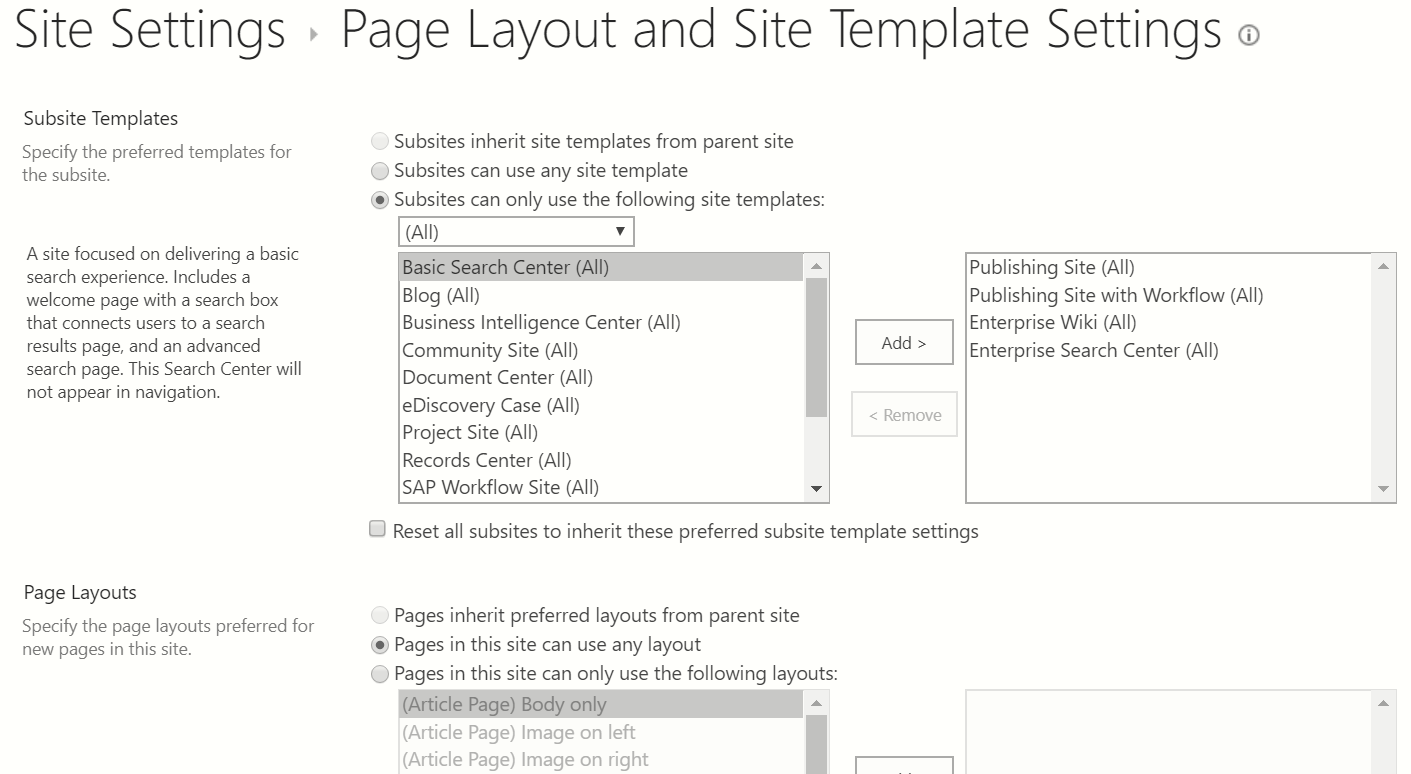
**Add the new page:**

**Site content=>page=>list=>new=>page**

****

**This uses to create our custom page.**

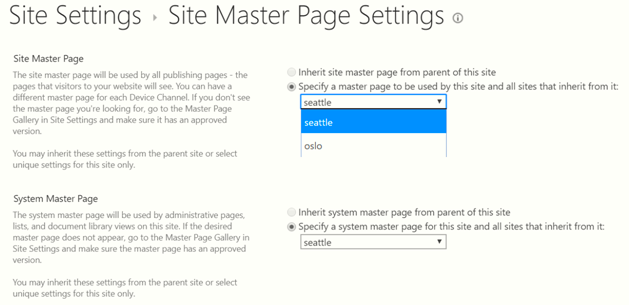
**Site setting=>**[Page layouts and site templates](https://nmek.sharepoint.com/sites/2019Publish/_layouts/15/AreaTemplateSettings.aspx) =>

****

**Master page**

**NB:it has 2 pages: seatle=>with left navigation and oslo=>without left navigation**

**but in publishing page we can add our own custom page in it. the results display on home page.**

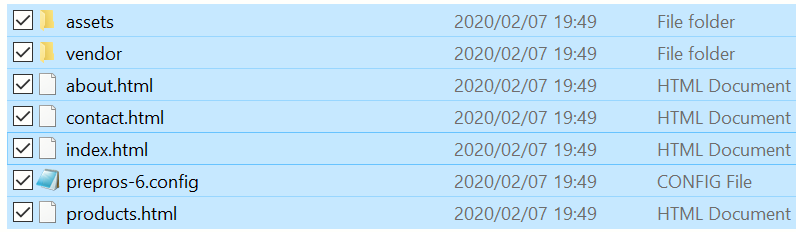
****

**How can we create our master page**

**We need Free download some sources b/se it is difficult to start from the scratch. So that we use**

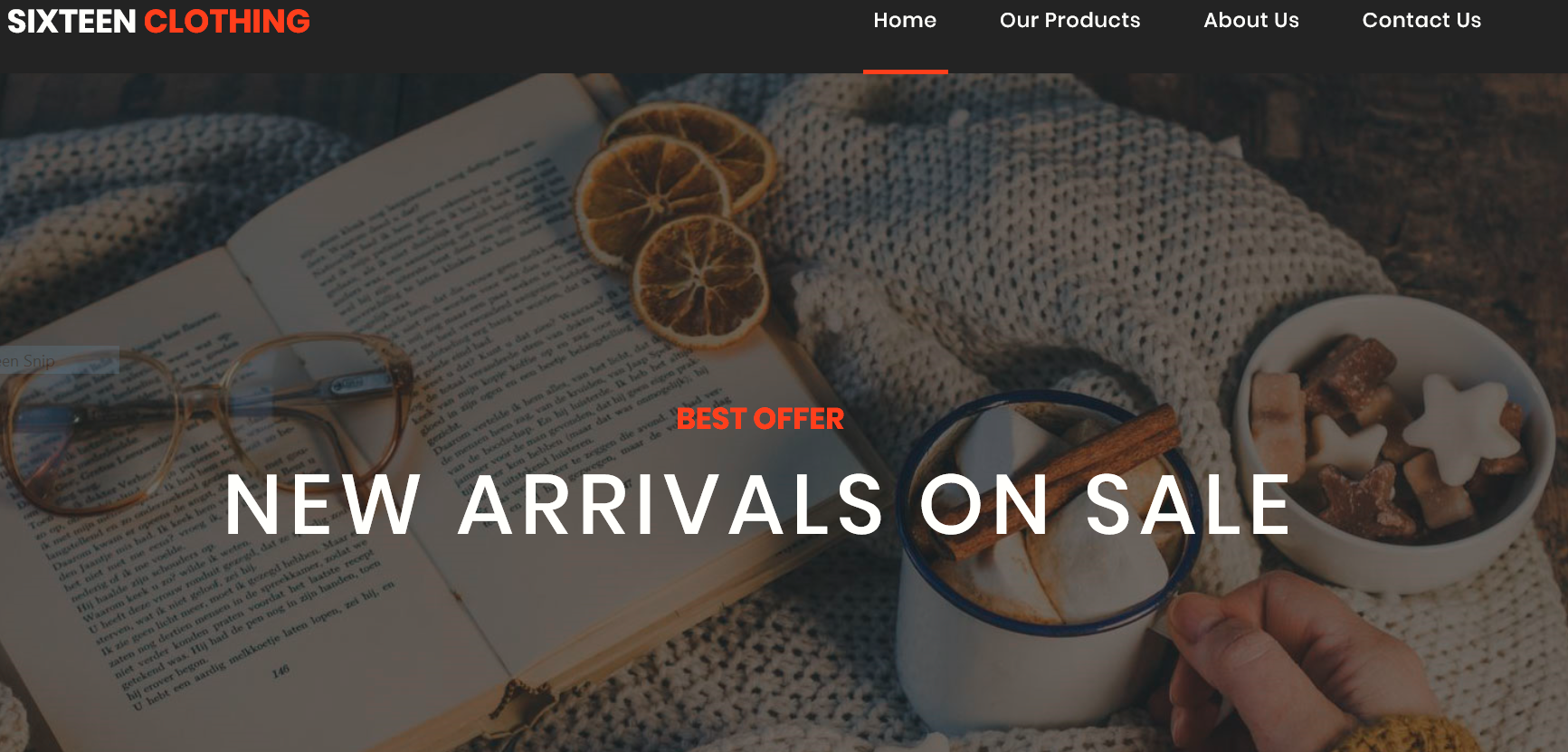
**Eg.**

* **Free Bootstrap 4 html css template then google it**
* **Free Bootstrap 4 Website Templates by TemplateMo**
* **templatemo.com/tag/bootstrap4 =>then we grape the one which we want.**
* **546 Sixteen Clothing**
* **Live view, if you want it**
* **Download to our workspace**
* **and then extract it, then we get**

****

**1.index.html:**

**In here we get a view, we get the others in this page.**

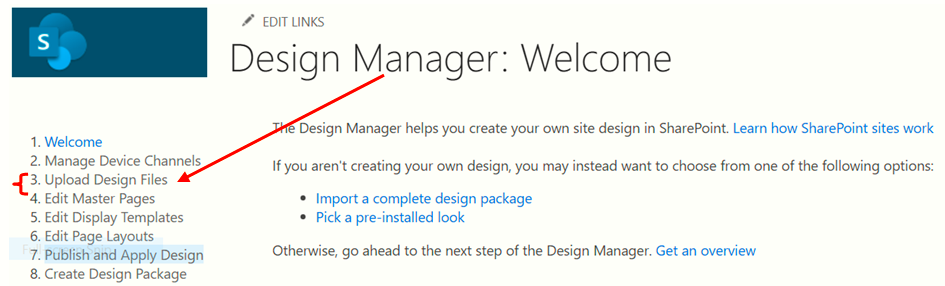
****

**Open it in visual code: look everything in index.html**

**Branding:**

**Design manager: used to create customizing but we do everything first.**

**Site setting=>Design manager=>**

****

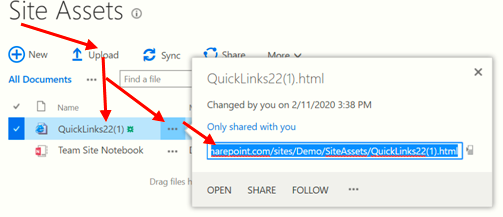
[Upload Design Files](https://nmek.sharepoint.com/sites/2019Publish/_layouts/15/DesignMapDrive.aspx)

* Upload the HTML version of your Master Page
* Upload CSS and js files and any other design assets
* Access design files for editing

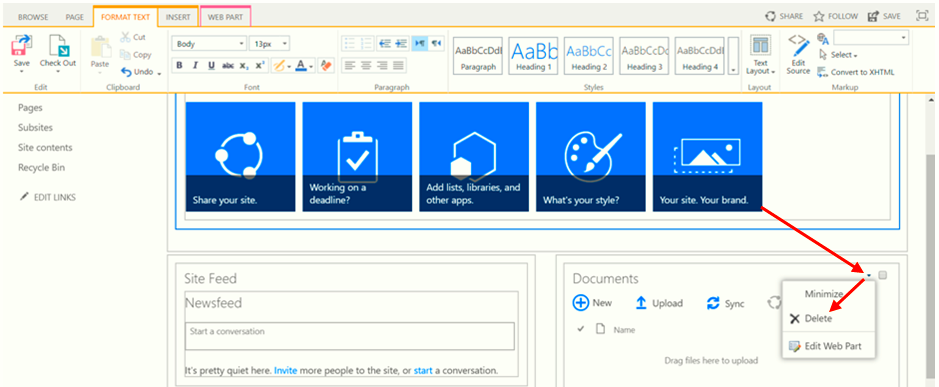
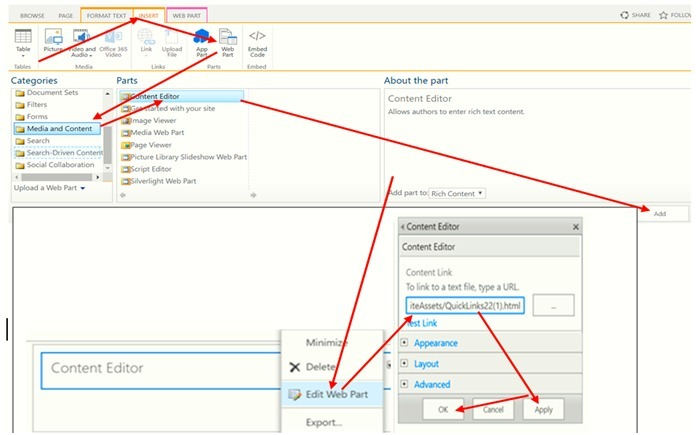
[Edit Master Pages](https://nmek.sharepoint.com/sites/2019Publish/_layouts/15/DesignMasterPages.aspx)

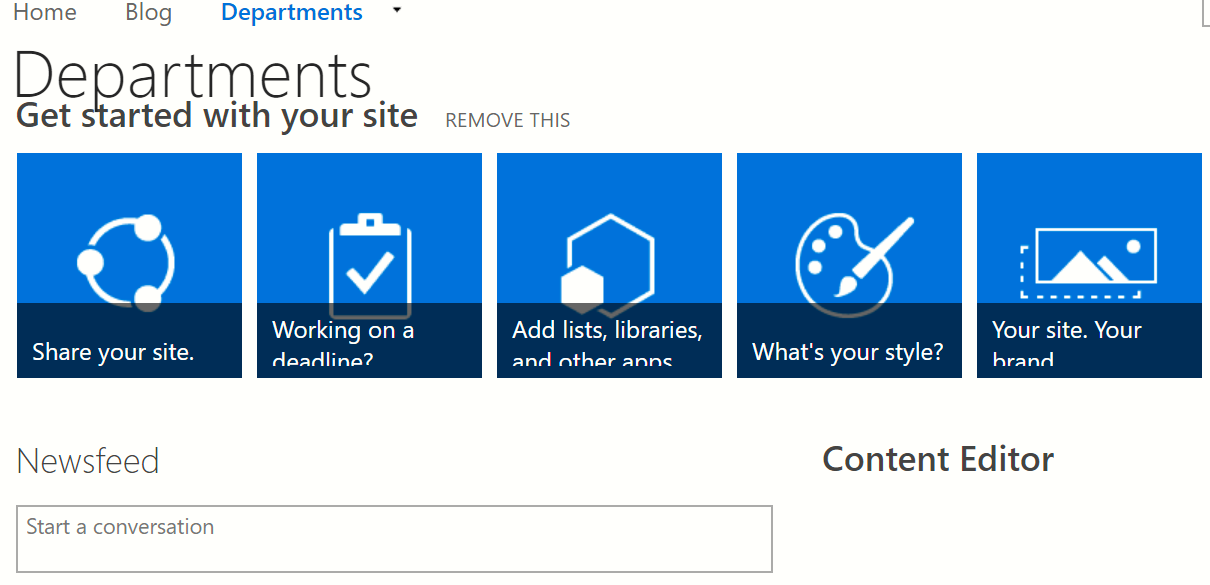
* [Convert an HTML file to a SharePoint master page](javascript:%7b%7d)
* [Create a minimal master page](https://nmek.sharepoint.com/sites/2019Publish/_layouts/15/CreateNewHtml.aspx?Mode=MP)

**bootstrap theme campany-w3schools**

****

**Copy the url and paste it at the page of Department**

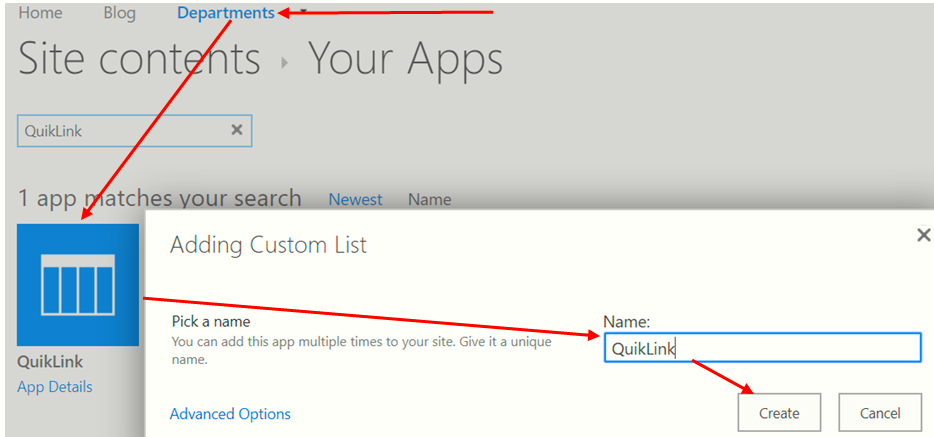
**** ****

****

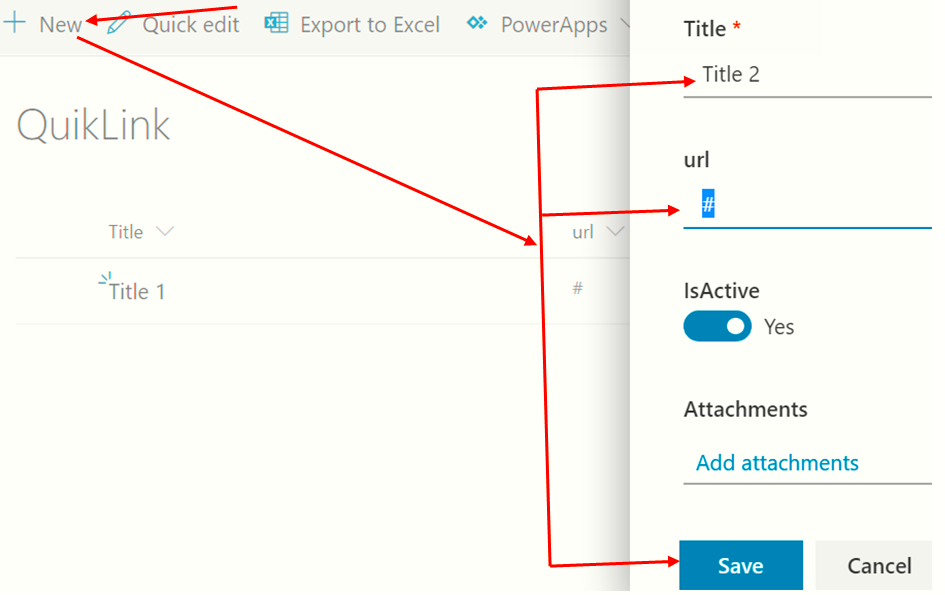
## It is applied but the content is not coming, look Content Editor

**To bring the content**

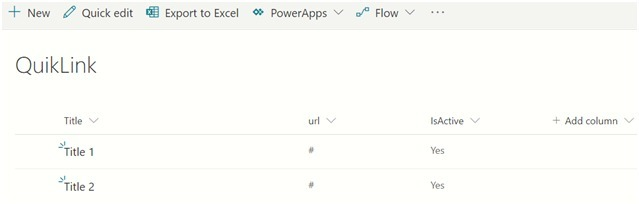
* 1. **Save QuikLink as atemplate=>under Home oage**
  2. **Add QuikLink as atemplate to our Department**

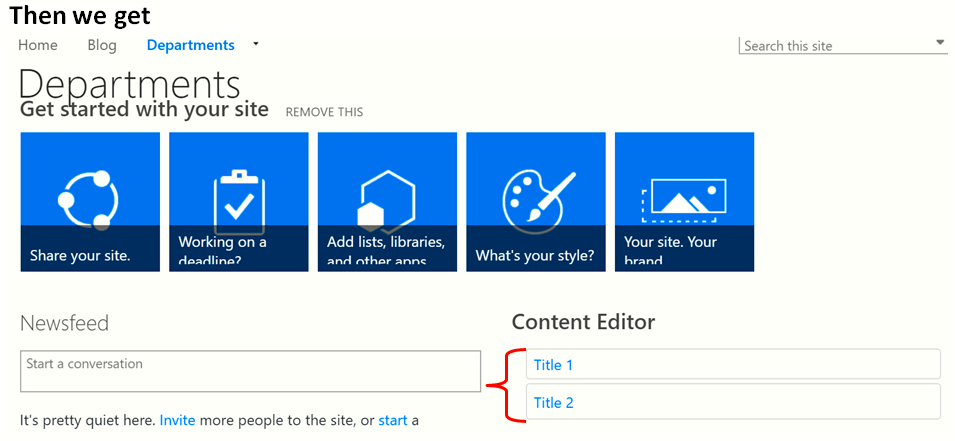
****

**Open the Template and add some items as follow**

****

**The we get**

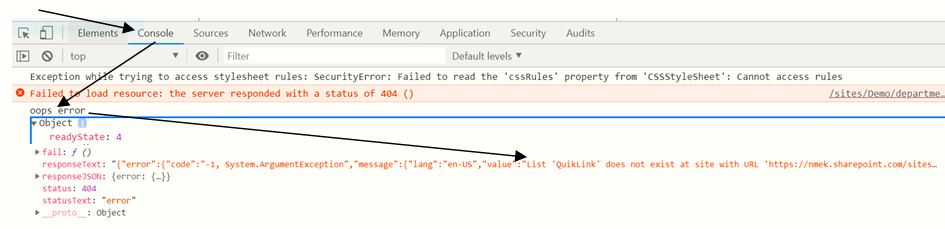
****

****

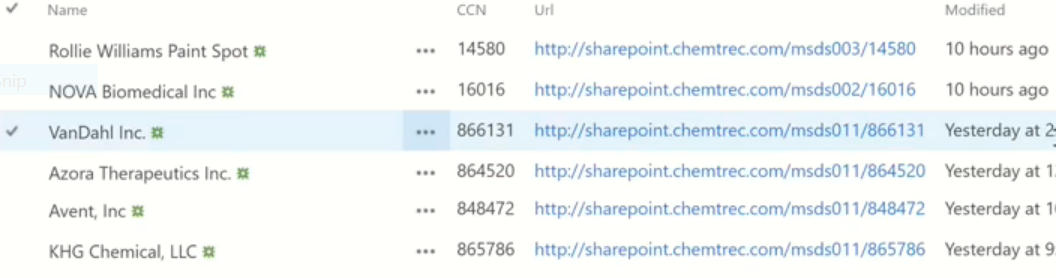
**NB:**If we want to applay the same link it HR . follow the same way by adding QuikLink Template in HR.

**Let me try with out: Adding the template**

**Go to Site Asset in Home =>copy url of QuikLink=>Paste it at Edit page of HR. then we get the error.**

****

**Creating one list to search all lists in site collection**

****

We do them by using HTML, CSS, RESTAPI and Bootstrap to do the code of this list

**Use Rest api(YARC) and then use the following next to your sharpoint url**

**/-api/web/Lists/GrtByTitle/('searchingList')/item?$select=Title,CCN,url& $top =999999**

If you want to talk about HR Department what do you talk? When the new Employee hired display picture’s of them to announce for the other. In the interanet.

**Project1:**

**To display the new hire**

**Look the following Template(team)**

Browse Employee html css

### [How To Create a Meet The Team Page - W3Schools](https://www.w3schools.com/howto/howto_css_team.asp)