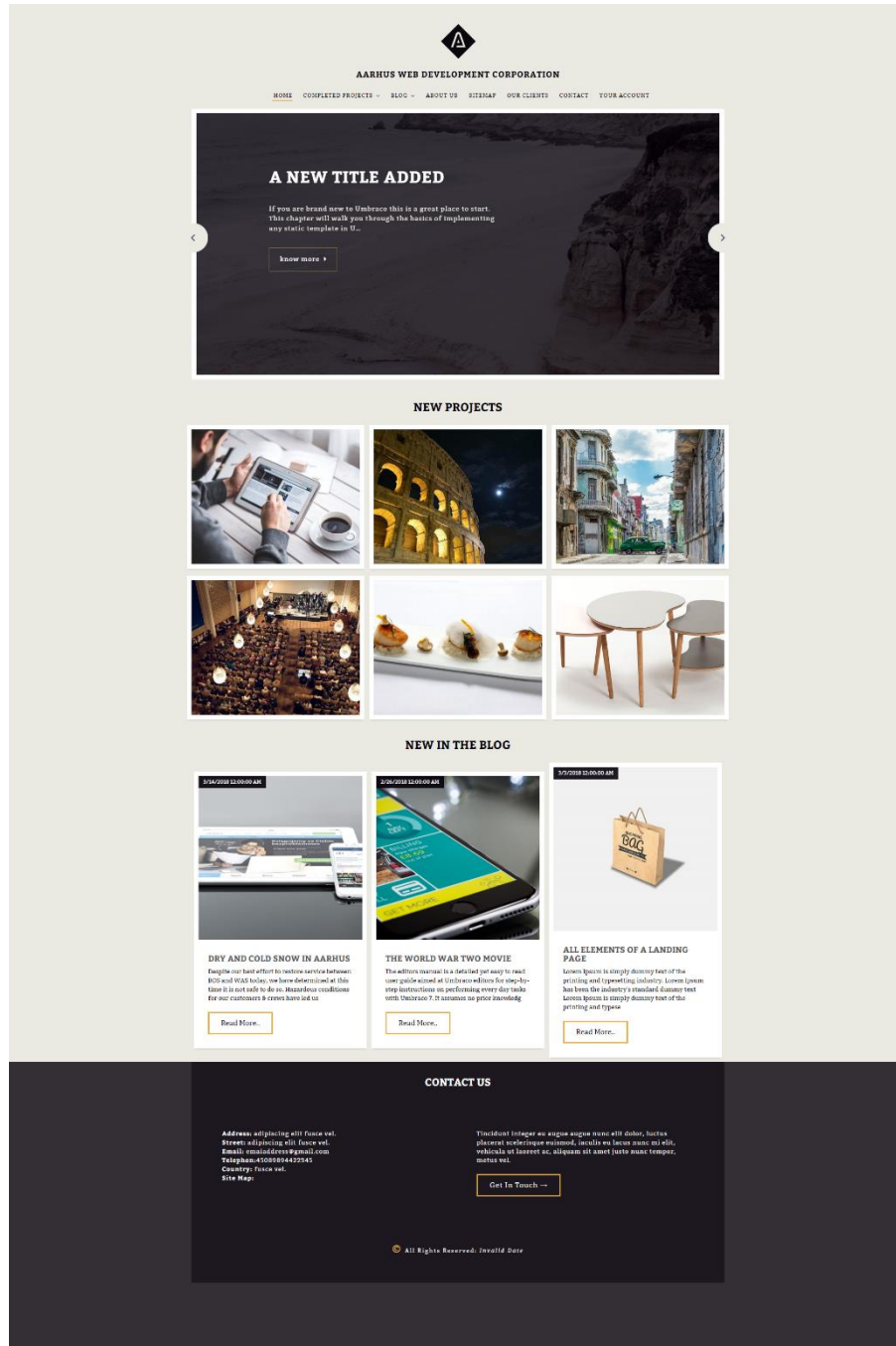


# Umbraco Assignment

26 March 24, 2018



Second Semester: Munir

## 1.Intro

Since I was very used to work with WordPress as Content Management System solution, it was very hard for me to make my head around it. The Templating, Doctypes and Compositions components were not easy for me to grasp and use. But checking out documentations, class materials and online resources make it possible.

I did my Umbraco installation from the console manager window, but I assigned self-defined / made database for it into SQL server. I created the database and then connecting it to the Umbraco installation inside visual studio.

## 2.Website Structure

Header section where website logo and title along with main Navigation is located. Inside Umbraco backoffice I have created compositions folder where I placed my compositions without any template attached to it, so I can use it on any page from the composition section.

In compositions I have HeaderProperties where Logo and title of the page is set and outputted. Menu NaviHide Property is also created as a composition and is assigned to different pages in the Main Menu. Lastly the SEO properties are also of type composition and is placed inside composition folder but is only used to master template for site description, keywords, meta data and site title.

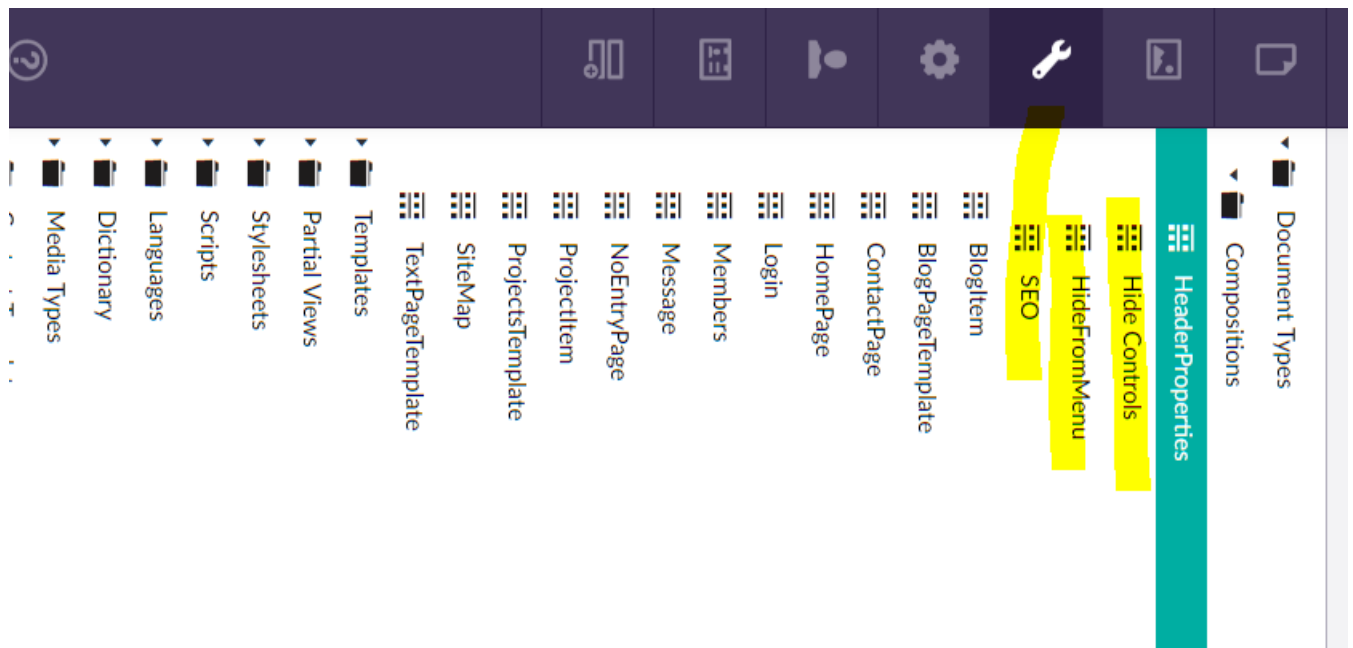


Figure 1: shows the templating structure of the back office.

### 3. Main Navigation

In the back office I have created partial view called topNavigation. That is where code for the navigation is written.

```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@*
//Variables that give access to menu roots for each level
*@
@{
    var root = Model.Content.Site();
}

@{
    //access to children
    var selection = root.Children.Where(x => x.IsVisible()).ToArray();

    var selectionChilds = Model.Content.AncestorOrSelf(1)
        .Children.Where(x => x.DocumentTypeAlias != "projectsTemplate")
        .First()
        .Children
        .Where(x => x.IsVisible());

    IEnumerable<IPublishedContent> projects = Umbraco.TypedContentAtRoot()
        .First() //Root document
        .Children() //Level 2 documents
        .Where(x => x.DocumentTypeAlias == "projectsTemplate")
        .First() //Projects overview document
        .Children() //Projects documents
        .Where(x => x.HasValue("projectStatus") && Umbraco.GetPreValueAsString(x.GetPropertyValue<int>("projectStatus")) == "Completed")
        .Where(x => x.IsVisible()); // Only visible documents
        //OrderByDescending(x => x.CreateDate); //Newest projects first

    IEnumerable<IPublishedContent> blogs = Umbraco.TypedContentAtRoot()
        .First() //Root document
        .Children() //Level 2 documents
        .Where(x => x.DocumentTypeAlias == "blogPageTemplate")
        .First() //Projects overview document
        .Children() //Projects documents
        .Where(x => x.HasValue("postStatus") && Umbraco.GetPreValueAsString(x.GetPropertyValue<int>("postStatus")) == "Published")
        .Where(x => x.IsVisible()); // Only visible documents
        //OrderByDescending(x => x.CreateDate); //Newest projects first

}

<div id="header_menu" class="header__nav">
    <div class="nav-toggle visible-xs">
        <span class="nav-toggle_label">Menu</span>
        <span class="nav-toggle_icon"></span>
    </div>
    <nav class="nav" role="navigation">
        <div class="nav--mobile">
            <ul id="main-menu" class="nav__list">

                <li class="nav__item" <Html.Raw(Model.Content.Id == root.Id ? " is-active" : "")>
                    <a href="#">Home</a>
                </li>
            </ul>
        </div>
    </nav>
</div>
```

Figur.2

I have created variables for different levels at the top of the Partial Page Top Navigation and then used the variables for the different area loops to output menu. Outputting Menu was a challenging task for me, after searching internet and many readings I have created it, but still it need some bug fixing and a full revision.

In menu section I have implemented the specification based on roles and projects “completed”. And for blogs “Published” blogs logics into the Umbraco backoffice. It means only published blogs and completed projects will be displayed in the blog. The users can then able to publish more blogs and more projects.

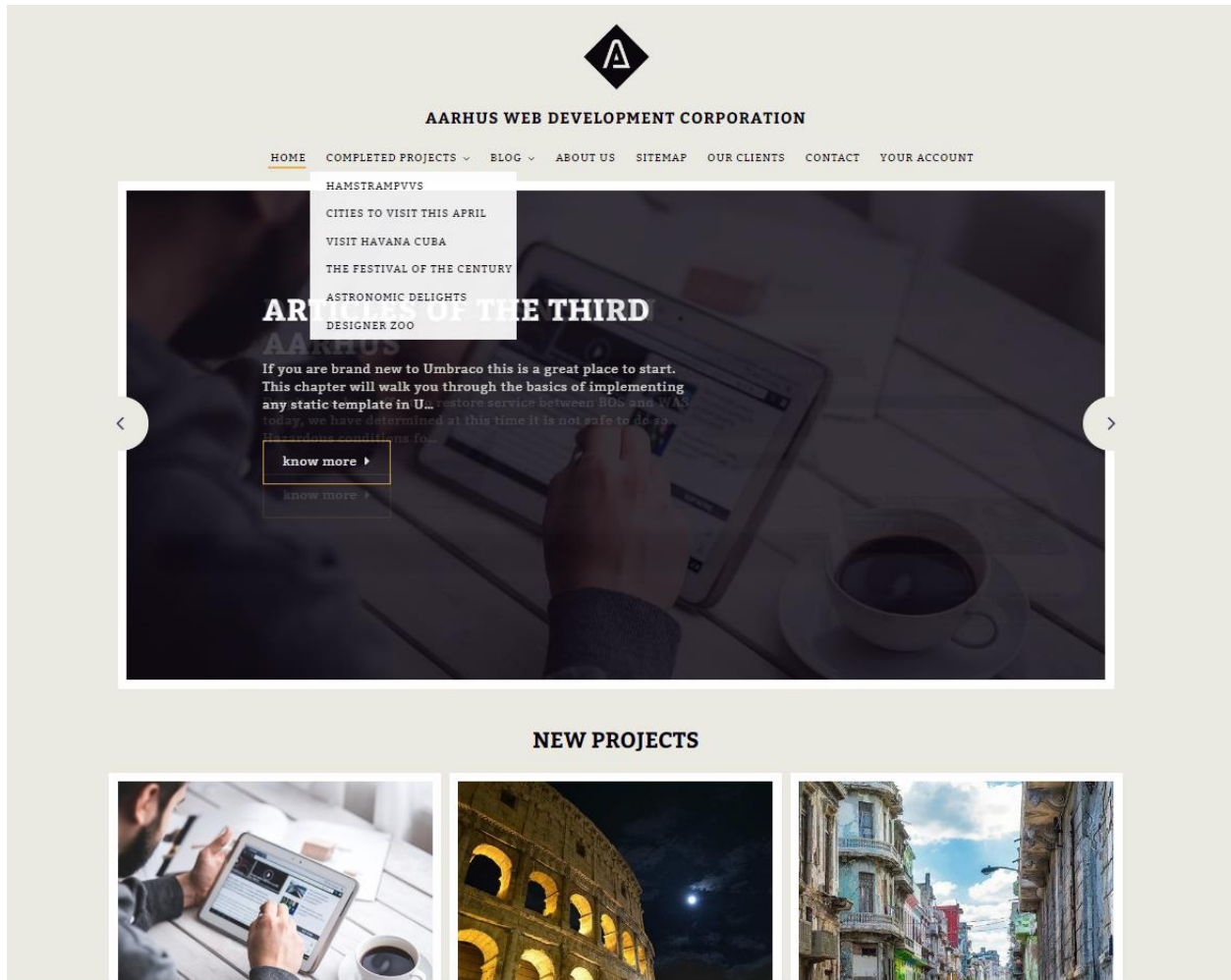


Figure-3

Figure 3 shows the feel and look of the website navigation and slider. The Navigation dropdown only displays the completed project and published blogs to the frontend of the webpage. The website is structured in way that it displays projects and blog posts to the Home page of the website.

## 4. Owl-Carousel Slider

Another important section is the Slider which is also created as a partial view and is used in MasterPage.

```
1 @inherits Umbraco.Web.Macros.PartialViewMacroPage
2
3
4 @
5
6 IEnumerable<IPublishedContent> newsItems = Umbraco.TypedContentAtRoot()
7 .First() //Root document
8 .Children() //Level 2 documents
9 .Where(x => x.DocumentTypeAlias == "blogPageTemplate")
10 .First() //Projects overview document
11 .Children() //Projects documents
12 .Where(x => x.HasValue("postStatus")) &&
13 Umbraco.GetPreValueAsString(x.GetPropertyValue<int>("postStatus")) == "Published";
14 }
15
16 @foreach (var child in newsItems)
17 {
18     <!--start-block-->
19     <div class="news-contain col-sm-12 col-md-4">
20         <div class="news-wrap">
21             <div class="news-wrap-top">
22                 <a class="news-wrap-top-link" href="@child.Url">
23
24                     @
25                     var BlogImage = Umbraco.Media(child.GetPropertyValue("blogImage")).ToString();
26                     if (BlogImage != null)
27                     {
28                         
29                     }
30                 </a>
31             </div>
32             <div class="news-wrap-body">
33                 <a href="@child.Url"></a><h2 class="news-wrap-body-title">@child.GetPropertyValue("blogTitle")</h2></a>
34                 <p class="news-wrap-body-text">@Umbraco.Truncate(child.GetPropertyValue("blogBodyText").ToString(), 190, false)</p>
35                 <a href="@child.Url" class="btn btn-carousel">Read More.</a>
36             </div>
37             <div class="news-wrap-date">@child.GetPropertyValue("blogDate")</div>
38         </div>
39     </div>
40     <!--end-block-->
41 }
42
```

Figure.4

In figure 3 the code shows the path to the level 3 item and displays the blog posts that are flagged “Published”. From the backoffice of the Umbraco you can choose how many posts to display on the frontend of the website.

The blog posts are then wired to the slider via Owl-Carousel slider, which is a Javascript slider library that is supported with HTML. The post image and the post title and description are outputted in the slider part of the page.

The Umbraco Truncate method is used on the description part of the blog posts where it subtract the text to 190 words per slide.

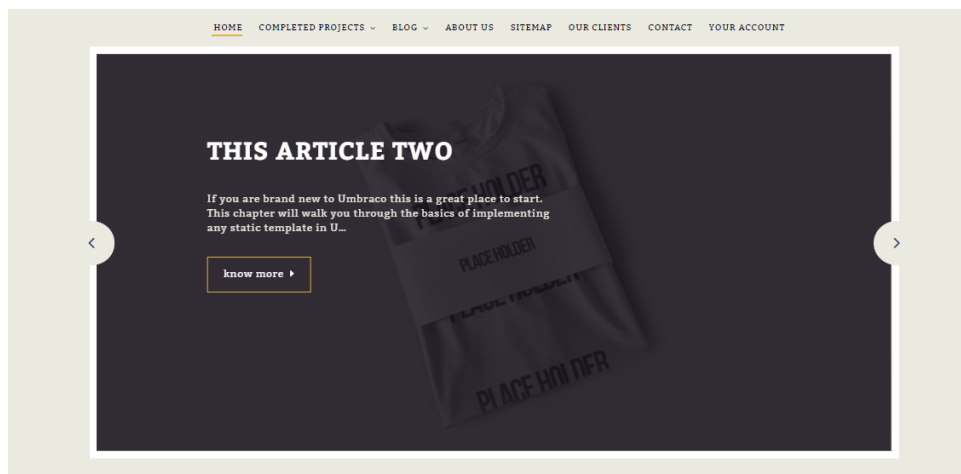


Figure-4

The blog slider images and text are outputted from a yet another partial view called blogSlider/ the blog slider executed contents from the blog posts, that means the first five sliders will be displayed into the slider.

## 5.Projects

In the projects section I have displayed the projects posts and those posts are then lead to a single project post display, for their detailed view. Projects are outputted again through a partial view. In the dropdown a list of only completed project appear and on the front page/ home page of the website and on the all projects section all projects with a picture and some truncate text with a link will show up and that follows to a next full feature of the project post.

On projects role based public access has been applied and that means if the Active member clicks of the project named confidential then the NO ENTRY page will appear.

The Active member will also not see some other menu Items which are protect from the user based on her/his role.

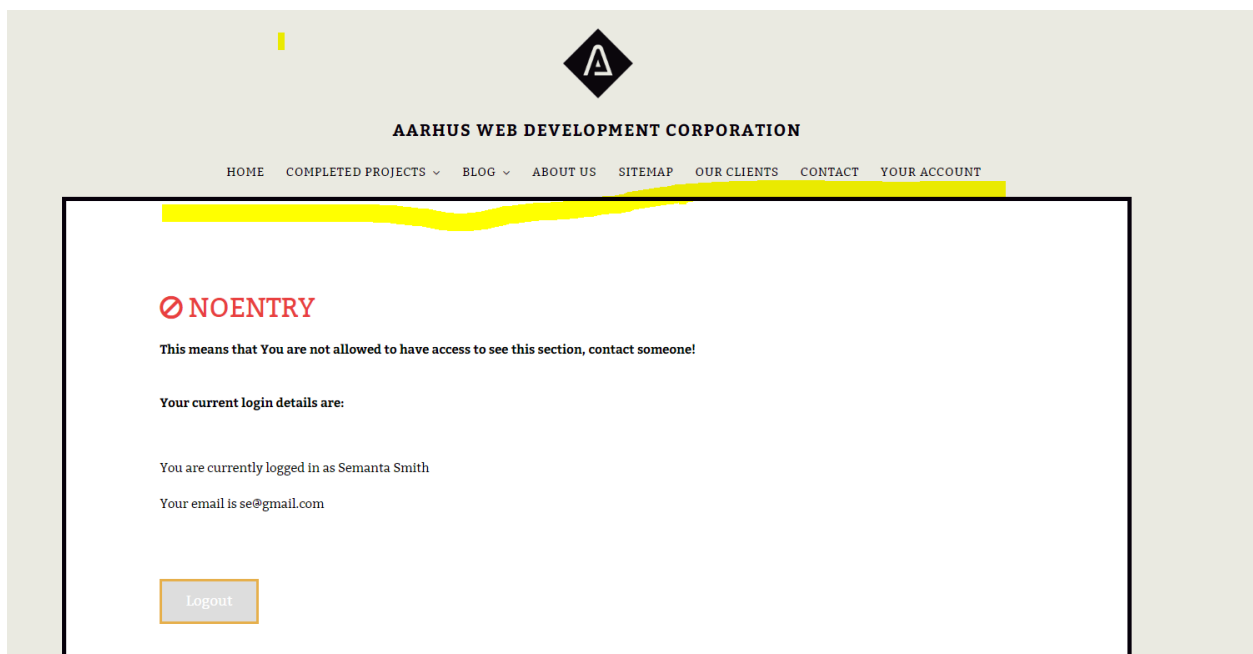


Figure-5

In the figure 5 the no Entry page has been shown to Semanta Smith with the email: [se@gmail.com](mailto:se@gmail.com)

And password: **semantasemanta**

## 6. Blog Posts

The blog posts are inserted through a Macro Partial View. And then it is available through Rich Text Editor and Grid.

Blog posts are also used for the slider section where the sliders will appear based on how many posts are displayed in the blog section.

From the backoffice the user can add blog posts and remove them by clicking on radio buttons. If a blog post is published, then it will appear on the frontend of the website otherwise it will not.

## 7. Access

You can set role and user based access to the content of the website. If you are an active member the authoritative members, admins in most cases are able to change access to the specific section of the website.

## 8. SASS

The whole project is sassified, and I have created partial sass files for each component of the website. And then applied my sass styles which are then compiled to css and min.css files.

```
[
  {
    "outputFile": "Content/CSS/bootstrap.css",
    "inputFile": "Content/bootstrap/bootstrap.scss"
  },
  {
    "outputFile": "Content/CSS/customStyles.css",
    "inputFile": "Content/custom-styles/customStyles.scss"
  }
]
```

This compiler config file indicates how and where the files will be compiled to. I have designed the website with sass and it was fun to work with, I have added sass animations, mixins and functions to the project.

## 9. SurfaceController

A model with name, email, subject and message properties was added to the project root folder and then a controller was defined to send messages. The messages are then connected to the back office and can be displayed in the backoffice too.

I have created a partial view for the contactForm that displays the contact form to the contact page of the website.

## 10. Conclusion

Working with Umbraco and specially this project was a huge learning process. Since I used to work with WordPress a lot, at first it was not easy for me to understand the logic of making templates and doctypes. But then I found out that doctypes and templating is core to Umbraco. I had challenges to know the difference between Umbraco ViewPage and Umbraco Template Page. When I turned on the

Dll mode for the project then the model generating and referencing to those was not easing and it keep giving me errors, which I still need to work with and find out how to overcome the challenge.