Seminar 1 report

Internet Applications, ID1354

Niclas Fölster - nfolster@kth.se

Sun 6 November 2017

1 Introduction

This task involves creating a website for recipes. The requirements for the solution is the following. The website needs to be created in HTML and use CSS stylesheets. The website shall include one index page, one calendar page, and at least two recipe pages, and all pages must be able to reach each other. They shall be able to reach each other with a distinct and easy-to-use navigation link menu. The calendar page shall depict a calendar of a month with pictures in at least two days of the month. These pictures shall contain links to the recipe on the picture.

Part of the task is also to start up a web server and put the website on it.

The website shall also look exactly the same in all browsers.

There are two more requirements for extra points. The first is that the website must be responsive, the second is that the site must follow the four accessibility guidelines listed below.

1. Use text alternatives.
2. Don’t rely on color alone.
3. Use HTML and CSS properly.
4. Provide clear navigation mechanisms

2 Literature Study

Since I am re-reading this course, I have both experience and knowledge from last year at my disposal, which I used. I have also used w3schools a lot.

3 Method

I used Visual Studio Code to code html and css. This editor has a plugin to detect some syntax errors, which is good. To start up my web server I used MAMP, because it is compatible with Mac OSX, which is what I used. I used the W3C validator to check my website. I also tried to apply all the 10 basic heuristics as much as possible, however some were more applicable than others. I decided to implement both optional tasks.

4 Result

All pages passed the W3C validation. I had to do some minor changes to the code for this, but nothing visually changed.

All had this end result:

Document checking completed. No errors or warnings to show.

All my pages has a navigation bar, like below.

Illustration 1: The navigation bar

The navigation was created using an unordered list, with css to style it.

```
<nav class="menu">
    <ul>
        <li><a id="active" href="index.html"><img alt="Home" src="resources/home_active.png"><p>Home</p></a
        <a href="meatballs.html"><img  alt="Recipes" src="resources/recipe.png"><p>Recipes</p></a></li><li>
        <a href="calendar.html"><img  alt="Calendar" src="resources/calendar.png"><p>Calendar</p></a></li>
    </ul>
</nav>
```

Illustration 2: navigation bar html code.

The id “active” is used to show which website you are currently on.

I have used the class element in most things on my website to define how it shall be styled. As a result, for example, only a list with the id “menu” would be affected by the id “menu” attributes. To make it more stylish and informative I used commands as a:hover for example to change the color of the menu when you hover your cursor over it etc.

```css
/* ----------MENU---------- */
nav.menu ul {
    font-family:Verdana, Geneva, sans-serif;
    font-size: 25px;
    margin-left: 0;
    margin-top: 0;
    text-align: center;
    position: fixed;
    background-color: rgb(88, 196, 140);
    overflow: hidden;
    width: 100%;
}
nav.menu li{
    display: inline-block;
}
nav.menu li a{
    margin-top: 0;
    padding-top: 7px;
    padding-bottom: 5px;
    padding-left: 2px;
    padding-right: 10px;
    display: block;
    color: white;
    text-decoration: none;
}
nav.menu li a:hover{
    /*background-color: rgb(128, 236, 180);*/
    background-color: rgb(68, 176, 120);
}
nav.menu li a#active{
    color: rgb(88, 196, 140);
    background-color: white;
}
nav.menu img{
    width: 25px;
    padding-right:5px;
    padding-left: 10px;
}
```

Illustration 3: CSS code of the navigation bar

The menu appears on all pages, so all are connected.

Clicking on the Recipe page on the main navigation bar leads to the first recipe. On all recipe pages there is a second navigation bar to go between recipes.

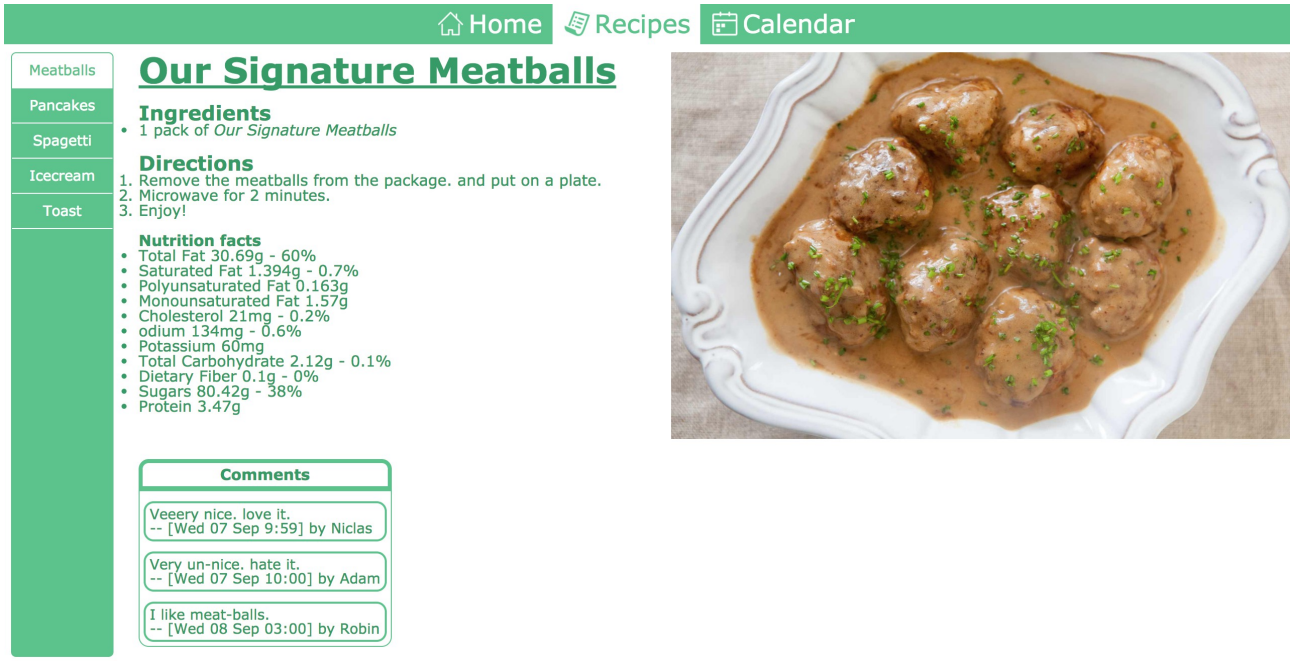

Illustration 4: The Recipes page.

The calendar is created with a large table containing 30 cells. Where the pictures are links to the recipe pages, and the green cell is the current day(that is hard-coded though).

Illustration 5: The Calendar page.

The page looks the same in all browsers, since I'm using the reset.css document provided. The above illustrations are from chrome. Here is a picture from safari.

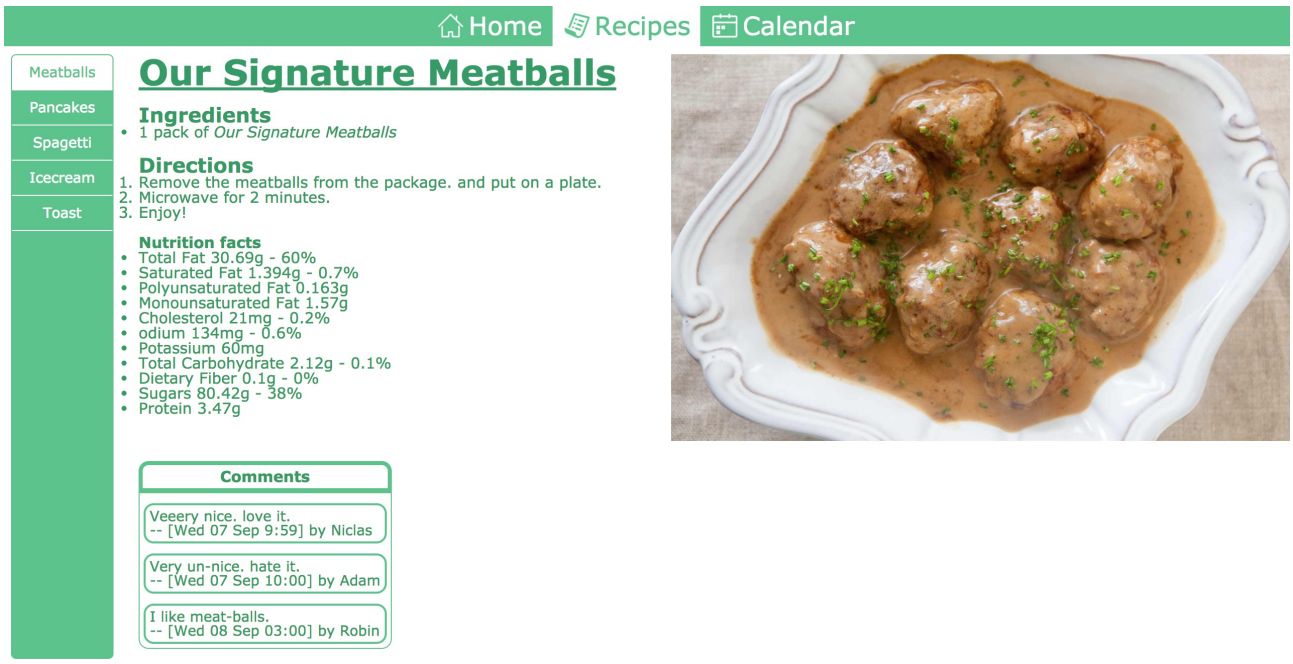

Illustration 6: Recipes opened in Safari.

When trying to implement the 10 basic heuristics for user interface design, the result was as follows.

- **Visibility of system status**
  - This one I think was not very applicable in this early stage of the website, except for the hover function, which shows realtime when the mouse hovers over a link.
- **Match between system and the real world**
  - This one is more applicable, and also mostly applied naturally. I use no system-oriented terms that shows on the website. Everything consists of concepts and phrases familiar to the user, no matter the extent of knowledge about html etc.
- **Consistency and standards**
  - This is mostly applied with aid from the reset.css, so the user will get the same experience no matter what browser they enter the website through.
- **Recognition rather than recall**
  - All objects on the page are always visible. There isn't much to do here. All information about one recipe is of course on the same page, so the user doesn't have to switch pages and remember earlier information. With help from the navigation bar, and the extra navigation in the recipes-page, the user always has full freedom and comfort to navigate through the website, without ever getting the feeling of a “dead end”. This makes all information easily accessible.
- **Aesthetic and minimalist design**
  - This is a very important part, I believe. A simple and clean design on a website can prove to be much more efficient, clear, and easy to use. This is also applicable in design choices, like colors, and different slide down bars etc. To have everything on the same

page, where you have to scroll a lot etc. The use of a simple navigation bar that exists on every page prevents much confusion I think.

The four accessibility guidelines is implemented as follows.

- Use text alternatives.

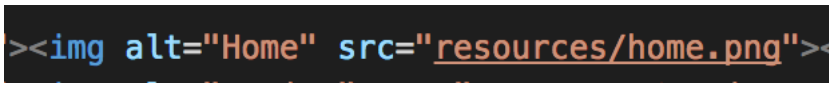

Illustration 8: The use of "alt" to give text alternatives to images

By using the "alt" attribute in the "img" tag, this guideline is easily accomplished. I also used icons in the navigation bar.

- Don't rely on color alone.

  I do have color on the website, but no color is necessary to navigate or understand any information. If you can't see colors, you would still be able to see the change in color when hovering over the links, because it just becomes a lighter shade. So the website is still fully functional without color.

- Use HTML and CSS properly.

  I control all my code with a stylesheet, and do not change for example font-size directly in the tags. I also use the <nav> tag for the navigation menu etc.

- Provide clear navigation mechanisms

  As I explained under the 6th basic heuristic: "*With help from the navigation bar, and the extra "next recipe" buttons at the bottom of the page, the user always has full freedom and comfort to navigate through the website, without ever getting the feeling of a "dead end". This makes all information easily accessible.* "

I used @media screen to change font-sizes etc to have a responsive design. Here are the two stages for the recipe-page when you lower resolution.

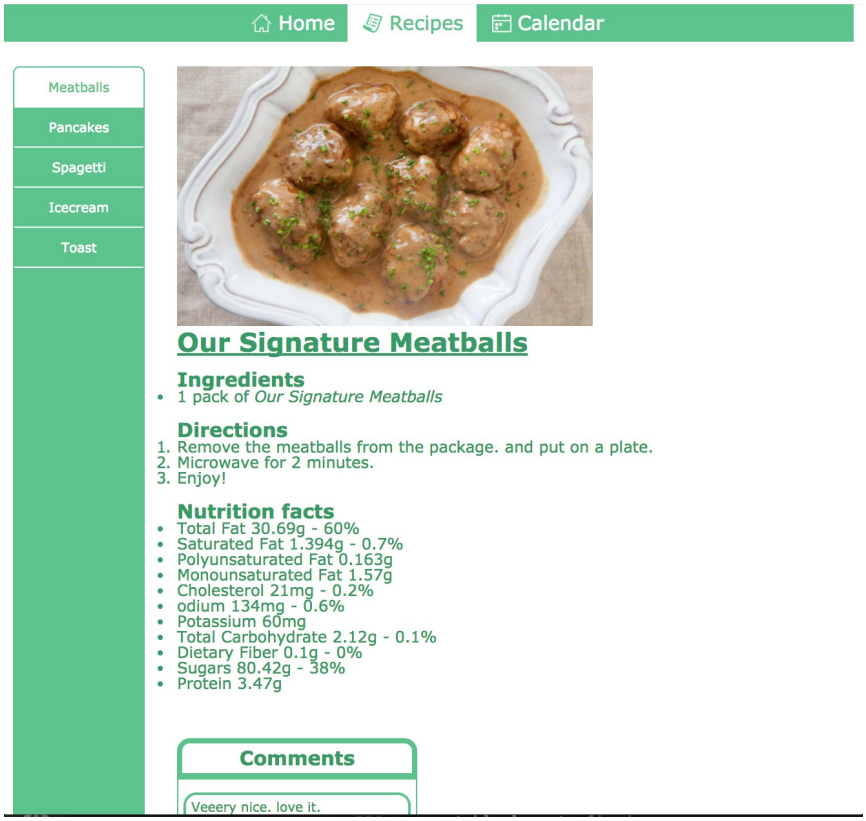

Illustration 9: lower resolution recipe-page.

Here the biggest change is font-sizes.

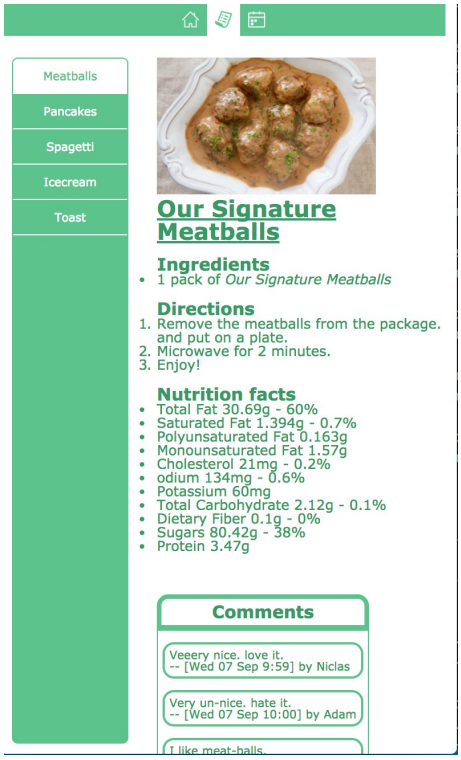

Illustration 10: lowest resolution recipe-page.

Here I have removed the text in the main navbar but kept the icons. The icons are so relatable that users can understand where they lead without the text.

Git link: https://github.com/nfolster/ID1354_Projekt

5 Discussion

I have fulfilled all the mandatory assignments, and all the optional tasks. I faced some design issues, but not much else. I used google and w3schools a lot. I had fun.

6 Comments About the Course

I will write this in Swedish. Jag gillar projektet med att göra en egen hemsida med relativt fria tyglar. Seminarierna är intressanta och givande därför att man får höra andras åsikter samt se hur andra har löst samma uppgift. Rapporten känner jag däremot gav mindre. Man skulle istället kunna ha redovisningstillfällen, där man visar sin kod och hemsida endast för läraren. Då kan man diskutera vad som gjorts bra, och vad som kunde gjorts bättre. Detta alltså utöver seminarierna, men istället för en rapport. Jag har lagt ner 10-12 timmar på denna uppgift, varav lite var för rapporten.