

# Jacobs tasty website!

Internet Applications, ID1354

Jacob Kimblad jacobki@kth.se

2018-11-05

## 1 Introduction

The first requirement for the tasty recipes web site is that it should consist of four different pages. An index page, a calendar page, a meatballs recipe page and a pancakes recipe page. The second requirement is that from each of the four web pages you should be able to navigate to any of the other three web pages directly. The third requirement is that they should all have a similar layout which should be put into consideration when choosing fonts, colors, mouse hovering, and link behaviour. In addition to all of the pages having similar layout some of their properties may not be assigned their default values. These properties are page layout, font family, font size, font style, foreground and background color, mouse hovering and link behavior. Beneath follows the requirements for each of the four individual pages.

### 1.1 Index page

The index page should be informative and welcoming by promoting the calendar page and providing a link to it in text-format.

### 1.2 Recipe pages

There should be an individual page for each of the two recipes (pancakes and meatballs). The recipe page should contain a picture of the prepared dish, its ingredients, instructions for preparing, its name and hard coded user comments.

### 1.3 Calendar page

The calendar page should, as the name suggests, present the user with a calendar of the month. Two days in the calendar should each contain a unique picture of the recipes (meatballs and pancakes). These should be clickable links that redirects the user to their respective recipe page.

## 1.4 Tasks

The tasks to be followed which should meet the requirements begin with creating HTML and CSS files to implement each of the different pages of the tasty recipes web site. It is necessary that all of the HTML and CSS files pass the W3C validation. It is also necessary to use a reset style sheet. When designing the website it is necessary to follow five of the ten heuristics for user interface design which were discussed during a lecture. All of the pages must also have identical behavior in the following four browsers: Chrome, Explorer (or Edge), Firefox and Safari.

## 1.5 Summary

The following bulle list summarises the requirements and functionality of the complete web page.

- Design and functionality:
  1. Contain index, calendar and recipe pages.
  2. Navigate to any page directly from any other page.
  3. Pages should use similar layout.
  4. Properties that must not have default values.
- Index page:
  1. Informing and welcoming.
  2. Contains a link to the calendar page.
- Recipe pages:
  1. One page for each recipe.
  2. Contain the name of the dish.
  3. Contain a picture of the prepared dish.
  4. Contain a list of required ingredients.
  5. Contain steps of how to prepare the dish.
  6. Contain comments from users about the dish.
- Calendar page:
  1. Present a calendar of one month.
  2. Present a picture of each recipe in two unique days.
  3. Make the pictures of the recipes clickable links to their respective page.
- Tasks:
  1. Implement the described pages using CSS and HTML.
  2. All CSS and HTML files should pass W3C validations.

3. Use a CSS reset sheet.
4. Follow the five given heuristics for user interface design.
5. All pages must have identical behaviour in the four common browsers.

## 2 Literature Study

For basic CSS and HTML syntax and tutorials reference *W3Schools Online Web Tutorials* [1] provides good overview of possible syntax for both languages. It also provides easy to understand examples and the limits and possibilities of different tags in HTML and selectors, properties and values in CSS. It also provides templates for websites using HTML and CSS which is a great resource to gather inspiration and examples of actual websites.

It is predicted that the hardest part to complete will be to get the formatting right for the calendar using HTML and CSS. The stackoverflow post shown in *html - What is this I hear about the table tag being deprecated?* [2] explains how the HTML table tag should not be used to format layout of the page. It should only be used for data that is very tabular in nature. When looking at the reference document for the table tag *HTML table tag* [3] we see that a lot of table attributes have been deprecated in HTML5. Reference *Using `div` Tags for Layout* [4] explains how using tables for layout is now officially wrong and instead div elements should be used. To help with constructing a layout for the calendar a div table generator such as *Div Based Table Generator* [5] can be used.

## 3 Method

To solve the tasks a set of development tools are selected either because the authors familiarity with them or because of their simplicity to use, both of which helps to speed up development time. The text editor used to edit the HTML and CSS files is Vim, described as "Vim is a highly configurable text editor for efficiently creating and changing any kind of text" *welcome home : vim online* [6]. The web server used is the http-server *http-server* [7] acquired under the NodeJS infrastructure. The shell used to run both Vim and the http-server is Zsh *ZSH - THE Z SHELL* [8] under a GNU/Linux operating system distribution known as Manjaro *Manjaro Linux* [9]. For source control Git was used with Github hosting a remote repository, this made it such that the validation tools, described in the next paragraph, could be used by just providing them with the URL's to the raw files hosted on Github.

Testing tools are also used to check some of the requirements of the web site are met. To make sure that the HTML files passes a W3C validation the online testing tool found in *The W3C Markup Validation Service* [10] is used. A similar testing tool is also used for the CSS files such that they also pass a W3C validation, this testing tool can be found at *The W3C CSS Validation Service* [11].

To help with debugging the code the web browser Firefox is used as it also includes a very handy set of tools, known as the Firefox Developer Tools *Firefox Developer Tools* [12], that can be used to change the source code and see the effects in real time all within the browser. This helps with speeding up development time, as there is no need to navigate around different tools to make and see the effects of a simple change. It is also necessary to test the finalised code such that it gives the exact desired behaviour in all of the different browsers defined by the requirements. To do this another online tool found at **online browser tester** was used.

## 4 Result

To make sure that every other page can be reached from any of the four pages a navigation bar at the top of each page is used. This navigation bar has the same look and layout for all four pages to fulfill the requirement of similar layout on all the pages and can be seen in figure 1.



Figure 1: A sample user interface screenshot to illustrate the navigation bar at the top of each page.

To get the same behaviour from all different browsers a reset style sheet was used which can be found in [13]. This was referenced from every single html file as to correctly reset the wanted css properties, the references can be seen at [14].

To make sure all pages have a similar layout a single custom stylesheet were used for all four of the pages which defines some custom properties of the page such as the fonts used, font sizes, font colors for headings [15][16] and paragraphs [17]. It also changes the default values for the foreground, background[18] and mouse hovering (seen in figure 1). The appropriate HTML tags and classes were then appointed to the desired sections to apply their respective layout.

Figure 2 shows a screenshot of the index page, which is the first page that is loaded when visiting the website. This page is both informative and welcoming while also providing a link to the calendar page in the text. For the recipe page the pancake is shown as an example. Figure 3 shows the top of the pancake recipe page which includes the title of the dish as well as the picture of the prepared dish. Scrolling further down the page reveals the ingredients, instructions and comment section which are shown in figure 4. The final page is the calendar page which is displayed in figure 5. As per the requirements it shows the days for a complete month as well as the pictures of the two dishes in separate days. The pictures are also clickable links which was done by enclosing the whole div tag in an a tag.

To validate the HTML and CSS files the W3 validators *The W3C Markup Validation Service* [10] and *The W3C CSS Validation Service* [11] were used. The results are shown

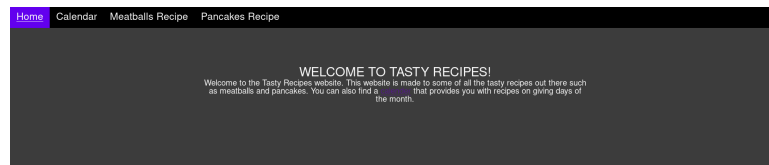


Figure 2: A sample user interface screenshot to illustrate the index page being welcoming and containing a link to the calendar page.

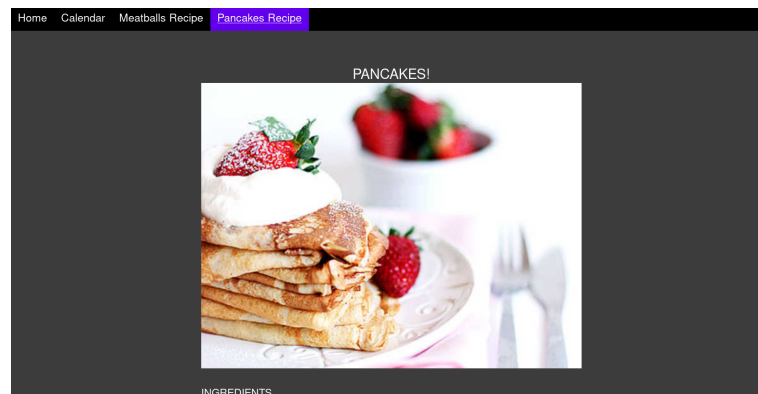


Figure 3: A sample user interface screenshot to illustrate the top of the pancake recipe page.

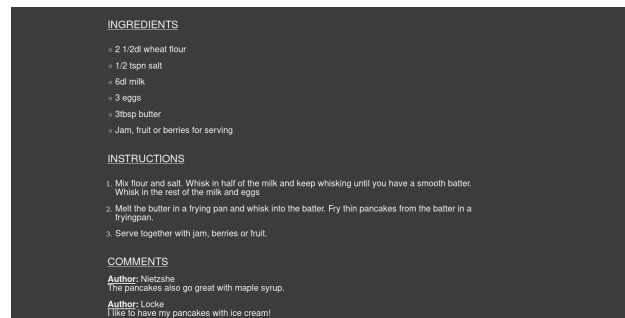


Figure 4: A sample user interface screenshot to illustrate the bottom of the pancake recipe page.

in figures 6-11.

The first heuristic which concerns the visibility of the system status has been taken into consideration when designing the navigation bar. To inform the user about what is going on the current page that is being visited is being colored in a different color in the navigation bar. In addition it is also underlined. This can be seen in figure 1. The

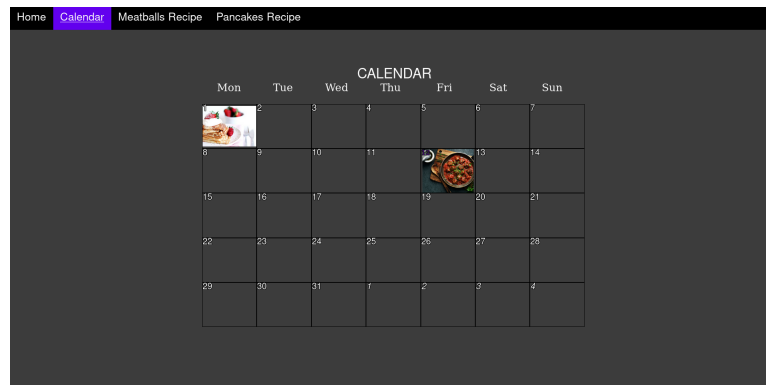


Figure 5: A sample user interface screenshot to illustrate the calendar page.



Figure 6: A screenshot showing the index page passing validation.

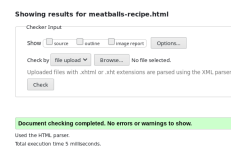


Figure 7: A screenshot showing the meatballs page passing validation.

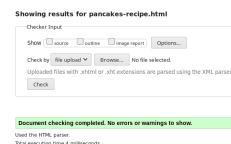


Figure 8: A screenshot showing the pancakes page passing validation.

second heuristics is taken into consideration when choosing names for the different pages in the navbar. Names that can easily be understood by the user by not being technical have been chosen which also can be seen in figure 1. In addition, the ingredients for the recipes are presented in a bullet list, and the instructions are presented in a numbered bullet list. Both of these should be very intuitive to the user and require no further

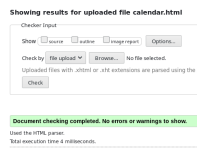


Figure 9: A screenshot showing the calendar page passing validation.

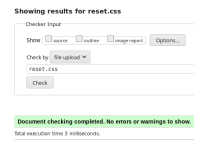


Figure 10: A screenshot showing the reset CSS passing validation.

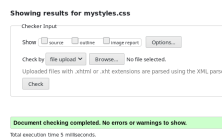


Figure 11: A screenshot showing the mystyles CSS passing validation.

explaining. These can be seen in figure 4. The consistency and standards heuristic has been put into consideration when navigating between different pages. Instead of learning new ways of navigation a very common navigation bar has been put at the top of the page easily available for the user. When designing the navbar the sixth heuristic has been taken into consideration such that the user can easily recognise how the navbar is used rather than try to recall some complicated instructions from memory. This is done by highlighting the current page and also highlighting when the mouse hovers over a different page as to show that the user is about to commit to a page switch if they press that specific button. The minimalist and aesthetic design has been put into consideration when choosing the colors for the page and when displaying information. To not reinvent the wheel, aesthetic colors have been chosen from the Material Design framework *Design - Material Design* [19]. A minimalist design has been presented by not overwhelming the user with information. A single navbar is used to navigate the entirety of the website and all the necessary information is gathered under a single page instead of splitting it up into several. The flow of the individual pages follow a very intuitive pattern.

As the tools suggested for testing the website in different browser (browsershots.org and turbo.net) doesn't allow for testing of local versions of websites (not even using localhost), the author was unable to test the website for other browsers than Chrome and Firefox, being that these two are the only ones who run natively on GNU/Linux.

However, figure 12 shows the calendar page running in chrome. The calendar page was chosen as demonstration as it contains much the same formatting as the other pages and more. We can see from the screenshot that the webpage is displayed precisely the same way as the previous screenshot which was taken in firefox making it such that the requirement is met.



Figure 12: A sample user interface screenshot to illustrate the calendar page running in chrome.

## 5 Discussion

Here follows a summary of all of the requirements formatted as a bullet list and a check next to each requirement that has been shown to be met earlier in the report.

- Design and functionality. ✓
- Index page. ✓
- Recipe pages. ✓
- Calendar page. ✓
- Tasks. ✓
  - Pass W3 validations. ✓
  - Use a reset sheet. ✓
  - Use interface design heuristics. ✓
  - Identical behaviour in different browsers. ✓

The most valuable lesson has been how to use CSS to format HTML divs into a grid network of desired look. This is valuable as the knowledge can be extended to format whole webpages using divs and not just something as specific as calendars. The problem that took the longest was getting borders around the individual dates in the calendar. A problem that was shown to be caused by faulty property Summarize the requirements



and *clearly state which of them you have met*. What lessons have you learned and what problems did you face? How were the problems solved? Should you have done something differently?

## 6 Comments About the Course

The author has probably spent around 20 hours on this assignment. Worth noting is that the author did not attend any of the first three lectures. Other thoughts is that the assignment is a very good challenge to get started and understand the interplay between HTML and CSS and makes the student look forward to the next assignment. Unfortunately the student came into the course late, which also was the reason for missing the first three lectures. The student thinks it would be interesting to see how the assignment would play out after attending the lectures as less time would probably have to be spent on getting started with the essentials.

## References

- [1] *W3schools Online Web Tutorials*. [Online]. Available: <https://www.w3schools.com/> (visited on 11/05/2018).
- [2] *Html - What is this I hear about the table tag being deprecated?* [Online]. Available: <https://stackoverflow.com/questions/3176896/what-is-this-i-hear-about-the-table-tag-being-deprecated> (visited on 11/05/2018).
- [3] *HTML table tag*. [Online]. Available: [https://www.w3schools.com/tags/tag\\_table.asp](https://www.w3schools.com/tags/tag_table.asp) (visited on 11/05/2018).
- [4] *Using `<div>` Tags for Layout*. [Online]. Available: <https://www.beginnersguidetohtml.com/guides/css/layout/div-tags> (visited on 11/05/2018).
- [5] *Div Based Table Generator*. [Online]. Available: <http://shabirhakim.net/Tools/divbasedtablegenerator.aspx> (visited on 11/05/2018).
- [6] *Welcome home : Vim online*. [Online]. Available: <https://www.vim.org/> (visited on 11/05/2018).
- [7] *Http-server*. [Online]. Available: <https://www.npmjs.com/package/http-server> (visited on 11/05/2018).
- [8] *ZSH - THE Z SHELL*. [Online]. Available: <http://zsh.sourceforge.net/> (visited on 11/05/2018).
- [9] *Manjaro Linux*. [Online]. Available: <https://manjaro.org/> (visited on 11/05/2018).
- [10] *The W3c Markup Validation Service*. [Online]. Available: <http://validator.w3.org/> (visited on 11/05/2018).
- [11] *The W3c CSS Validation Service*. [Online]. Available: <http://jigsaw.w3.org/css-validator/> (visited on 11/05/2018).

- 
- [12] *Firefox Developer Tools*, en-US. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Tools> (visited on 11/05/2018).
  - [13] original-date: 2018-11-03T11:46:12Z, Nov. 5, 2018. [Online]. Available: <https://github.com/jkimblad/ID1354/blob/master/seminar1/app/css/reset.css> (visited on 11/05/2018).
  - [14] original-date: 2018-11-03T11:46:12Z, Nov. 5, 2018. [Online]. Available: <https://github.com/jkimblad/ID1354/blob/master/seminar1/app/index.html#L8> (visited on 11/05/2018).
  - [15] original-date: 2018-11-03T11:46:12Z, Nov. 5, 2018. [Online]. Available: <https://github.com/jkimblad/ID1354/blob/master/seminar1/app/css/mystyles.css#L53> (visited on 11/05/2018).
  - [16] original-date: 2018-11-03T11:46:12Z, Nov. 5, 2018. [Online]. Available: <https://github.com/jkimblad/ID1354/blob/master/seminar1/app/css/mystyles.css#L62> (visited on 11/05/2018).
  - [17] original-date: 2018-11-03T11:46:12Z, Nov. 5, 2018. [Online]. Available: <https://github.com/jkimblad/ID1354/blob/master/seminar1/app/css/mystyles.css#L80> (visited on 11/05/2018).
  - [18] original-date: 2018-11-03T11:46:12Z, Nov. 5, 2018. [Online]. Available: <https://github.com/jkimblad/ID1354/blob/master/seminar1/app/css/mystyles.css#L50> (visited on 11/05/2018).
  - [19] (). Design - material design, [Online]. Available: <https://material.io/design/> (visited on 11/06/2018).