Readme

Alcomist

General

Webpage: http://r.nixo.no/web

We made a webpage where you can track your blood alcohol content (BAC) in 'real-time', during a night out. The way this works is by letting the website know when and what you drink and watch as the site provides an estimate for your BAC at the present time. We wanted a simplistic, beautiful and easy-to-use website and most of our technical and design choices are based on this. This also makes it more convenient as a cross-platform website, which we are aiming for. The website consists of userpages, history pages (where you can sort a table of your drinking history), and some about pages (where we give the users some useful information about drinking, health and the webpage itself).

Technical documentation

We have made some changes in the .htaccess file, so that instead of having ugly looking URL's all over the page, the htaccess is using a rewrite rule, so that instead of r.nixo.no/web/index.php?usr=Nixo, its presented as r.nixo.no/web/nixo. We chose to change this, to give the page a better aesthetic.

PHP

We chose to use php files as "classes" you can include in other files. This is so you only need to make things like the navbar and HTML "head" section once, not for every page, as well as not having to use some bigger framework, or install special software to run the project.

The *index.php* file handles user registration, redirection to correct page, and is the page you first see when entering the site. We thought about having a normal login page, but thought it would be simpler for the user to simply visit their user page directly instead of having to log in (r.nixo.no/web/username_here). One can also log in through the combined register/login form.

MYSQL

We chose to use a MySQL database to keep track of all the drinking events and user info, because it is easy to use with PHP, and it ensures data is stored across user sessions. Databases also accesses your information very quick, even if we had massive amounts of data to search through.

JS

We used JavaScript to create an algorithm to calculate your BAC based on attributes such as weight and gender. Calculating blood alcohol content (BAC) is difficult, as each individual responds to alcohol differently. We used a standard way of calculating BAC, obtained from the Norwegian Institute of Public Health, and configured this to work with our 'real-time' calculation, which is one of the reasons the JS does all the calculations.

Most of the dynamic elements on the site use JS and JQuery to function, and interacts with the server to store events and data. To be able to send and receive data on the fly, without having to reload, we had to use JQuery (AJAX requests).

CSS

We've taken advantage of the Bootstrap CSS library as a way to be more compatible with computers and devices of different sizes and pixels.
To keep the design up to date with the latest trends we chose a flat theme for the whole site.

Many elements on the main page (main.php) use new CSS3 functionality, like when you click a drink, it transforms rotation and shadow, with a transition. The color of the current BAC uses the CSS3 HSL color formatting, which allows for easy correlation between BAC and color. We used CSS when possible due to its better hardware acceleration then JS.

XML

We're using XML to save all the different drinks, so it is easily fetched from either the backend php, or frontend javascript. This makes it easy to add or remove what drinks we want on the page, and makes the maintaining a lot easier. In the XML, we're using one tag to keep track of the volume, and one for the displaying on the webpage itself. To fetch the XML data the PHP file reads the XML, processes it, and prints out the values for the drink buttons. The JS then uses an AJAX request to the server, which

converts the XML to JSON. This is because the JS had severe problems reading the XML, and all browsers except Chrome failed at opening it.

Input validation

Forms are used during login/user creation to establish the user's weight and gender, which are attributes needed to calculate the user's BAC. The user also has the option to change these attributes after user creation, through an identical form on the profile page. Form validation ensures that the user does not enter an invalid weight, which would be any number below 10 kg or above 635 kg . Radio buttons was the obvious choice for validation of gender, as you can only select one option.

Satisfying requirements

We wanted a clean, simple and playful design that worked with our "easy to use, nice to look at" vision where everything is stripped down to basic. No need to add more information/design/elements unless it advances the functionality. Therefore our drink-icons are 2D playful illustrations of the drink it represents, giving us the design-profile we're looking for.

We are using our drink-icon images from a website called Vecteezy[1]. The license is not explicitly Creative Commons, but it states "*You have permission to view, download, edit and remix this vector file or pack for personal and commercial purposes*" and demands a credit to Vecteezy if it's for commercial use. Most of our drinking-icons are as in the download, and some of the icons are redesigned by us to fit its drink better. Although our website site is not commercial (not yet anyway) we have chosen to include the credits in the "About Alcomist" page.

The site works in Chrome, FireFox, Internet Explorer, and most mobile browsers.

[1] http://www.vecteezy.com/vector-art/82082-free-flat-drink-vector-icon-set