Web Technology Report – Group 28



“Frendr”

*Bart Haak – 0899809*

*Jelle Kiers – 0846127*

*Duncan Scholten – 0903566*

*Maria Scholten - 0885441*

## General

Our web application is called “Frendr”. This is a platform for people to make friends. The application was inspired by Tinder, a dating application that matches people. “Frendr” allows the user to make a profile and add information about him/herself. Then, an algorithm is used to find the best match for that person. These people can interact with each other through the chat window that is provided on the website.

New users can make their own account (so this should be done when reviewing our assignment) and use the entered information to log in on the log in page. The user will then see his/her own personal page with matches, their interests and a chat room. Here, the user can interact with other users of the website and make new friends while doing so.

We mainly tested the website on Google Chrome, but also checked if it worked on both Internet Explorer and Mozilla Firefox. On all three, the website was responsive and worked as expected.

The nice thing about our web app is that it is very interactive. The user can keep changing his/her interests and new matches will appear when the user makes changes. The chat function is also very fun, and takes us back to the times when everyone used to used MSN. This makes the website very fun for the user to use and the website also has a social angle.

The user can chat with all other users in one large chat room. We thought it would be nice to also provide the possibility for everyone to interact with each other and not only with their matches. Sometimes, having the same interests does not result in people becoming friends while people who don’t share any interests might become the best of friends!

At the top of the page, the matches are displayed and the user can find his/her matches in the list of users on the right in the chat room. The user can change his name in the list on the right or chat anonymously. This makes the app extra special because it makes it easier to remain an anonymous user. However, if the user enters his/her own name, other users can recognize the name from their matches and start a private chat with their match. Then the real friend-making can start!

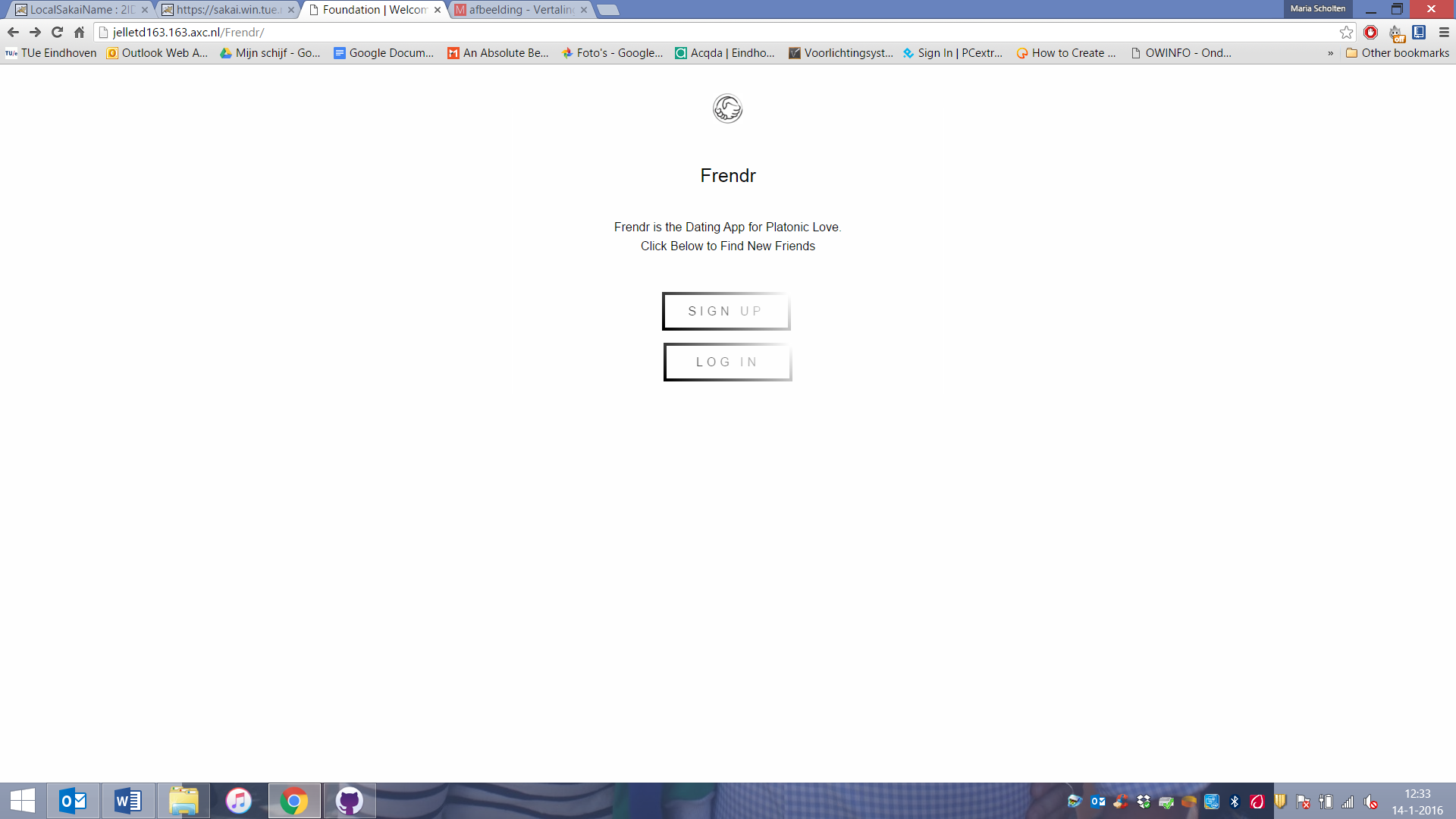
The website is hosted on …

## Frontend Design

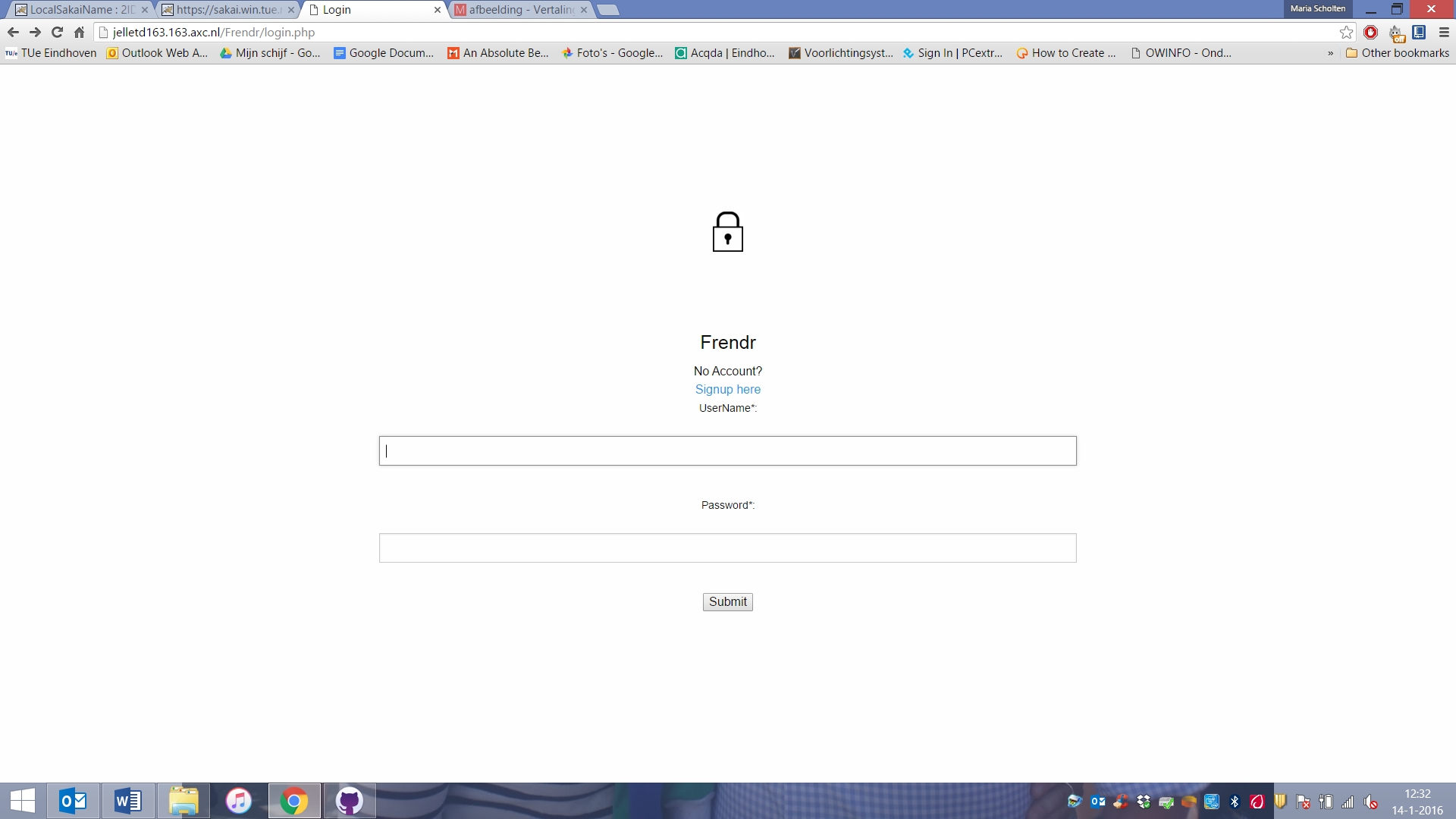
When first coming up with the design, we sat together and thought about all the things that we wanted for our website. Of course we wanted a Sign Up page, a Log In page and we wanted a page where the user could interact with other users. We wanted the website to feel intuitive and work in a way that would be fun for the user. Therefore, we decided to make one page with all the information in one place. On the left, a collapsible bar with information about the user itself would be shown, on top the matches could be found and at the bottom a chat room would be installed.

When designing this, we used two front-end frameworks. We used both Bootstrap and Foundation. We thought it would be fun to learn about another front-end framework so that’s why we decided to use Foundation as well. At the Login page, the user can use a button to login when he/she has entered the right information. However, when the user doesn’t have an account but does end up on the login page, an animation pops up where the user is directed to the Sign Up page. The same happens on the Sign up page. If the user does have an account, the animation directs the user to the Log in Page.

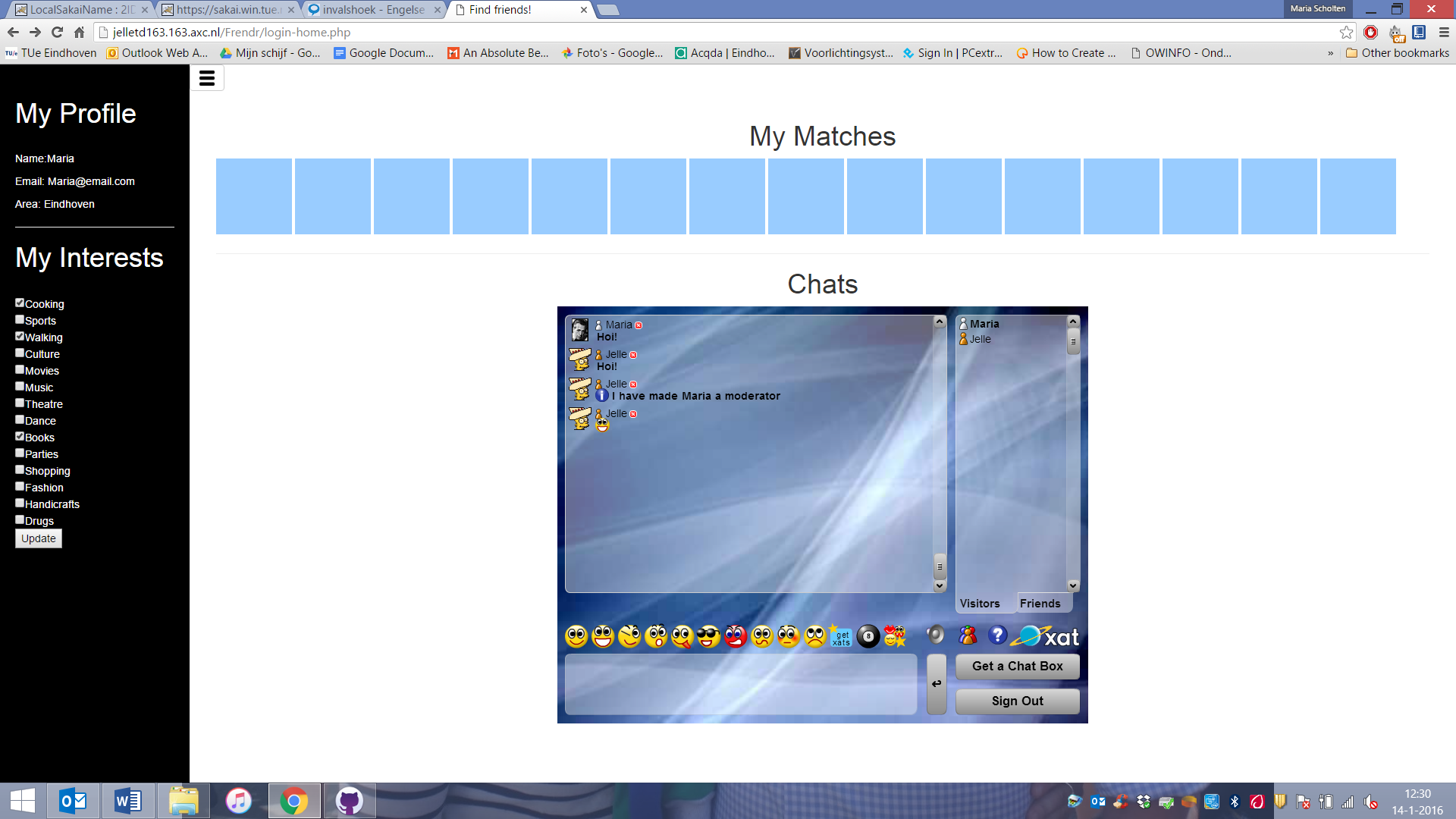
We use PHP and Javascript to handle the user interactions. The login and sign up is handled using both PHP and Javascript. We have used JQuery’s Javascript library. We used JQuery because it was easiest for us to write the difficult matching algorithm in Javascript (because we are most skilled at Javascript) and JQuery felt the most intuitive to use with our programming background. We also don’t have very difficult graphical requirements for our website, so there for using JQuery was the obvious choice.



*Figure 1: Home Page*



*Figure 2: Login Page*

**

*Figure 3: Profile page*

## Backend Design

Our group didn’t have much experience with web frameworks. Two people had used a bit of PHP before, and only one member had used Node.JS once before. This resulted in the fact that it was very difficult for us to set up a web framework. We tried working with Ruby on Rails as a web framework, but it turned out that this was very difficult for us and took up too much time to set up.

## Databases

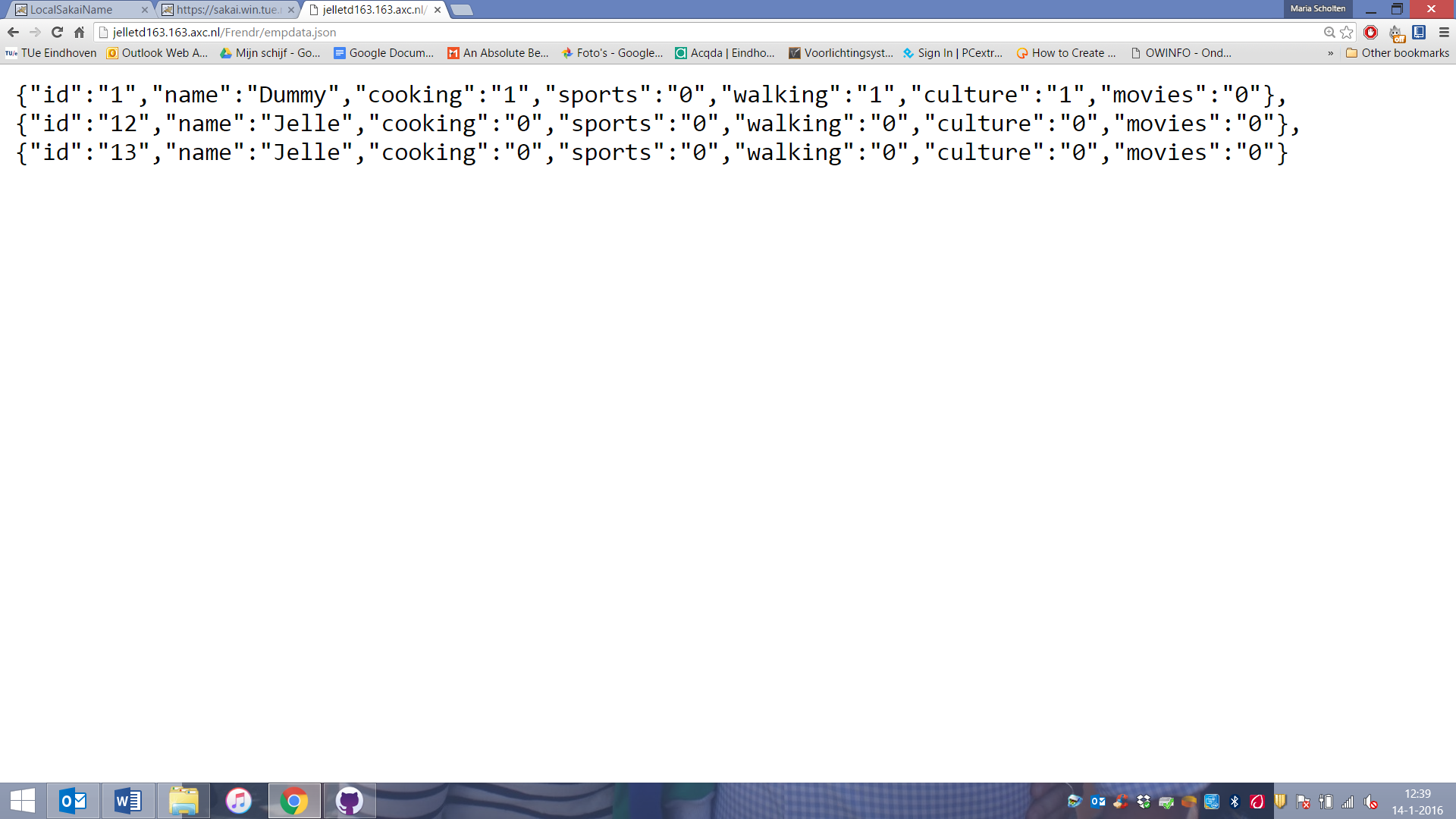
In our web app we store different kinds of information. We use one database with two different tables. In one table, user information is stored and in the other table the user’s interests can be found. We use an SQLite database to store the information. We use two different tables for easier data handling. The first table is used only to store the user’s information and the second table is used to store the interests that the user can enter later on the profile page. We use the second table to run the matching algorithm and find new matches for the user. Therefore it is more convenient to use two separate tables instead of one.

We connected the database using PHP. We use a standard connection between the website and the database. It is a matter of opening the connection and bringing in the information. On the Login page, the person that is logged in is stored in the query so we can provide personal information from the database on the next page (profile page).

The same is done when a person is creating an account. Information is then entered, the connection with the database is opened and the information is then stored in the database for later use (when logging in).

## REST APIs

On our website, we provide our own API. This API stores the user’s name and interests and we use this API to run the matching algorithm. The API is used to show which person has the highest percentage of similarities in interests with the current user. From the API, different information can be retrieved. The name of the user can be retrieved and the user’s interests. We provide the API in JSON and use JQuery to run the matching algorithm on this API.



*Figure 4: An example of the data that is provided in the API*

## Screencast

The link to the screencast of our website is:

In this screencast, you will be shown how the website works and how the user can interact with the website.