

Web Technology 2016-2017

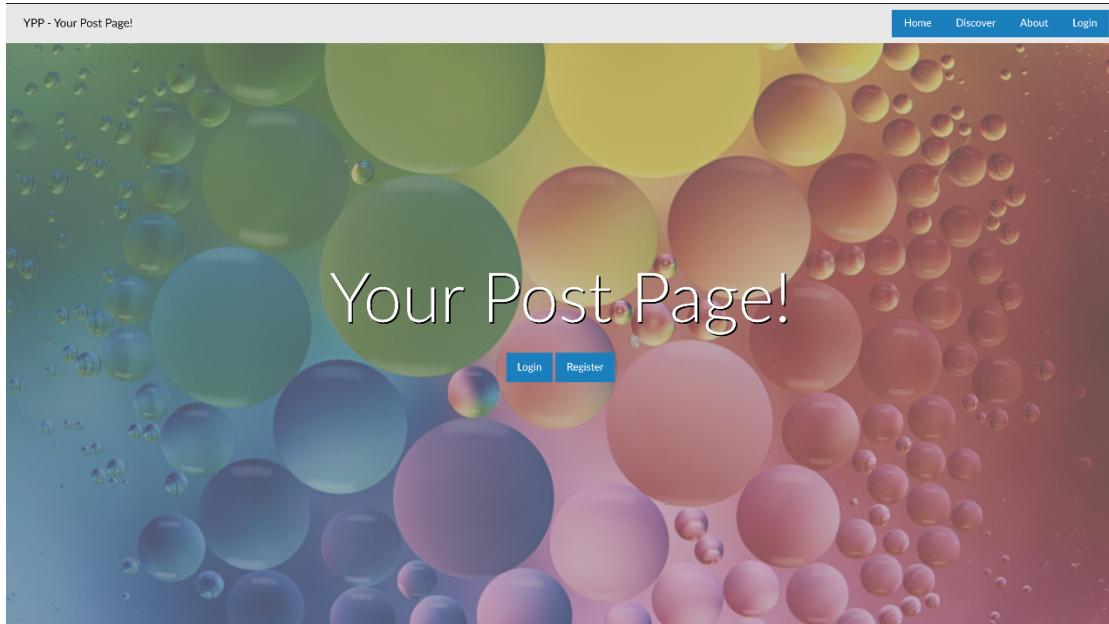
Final Assignment – Group 29: YPP - Your Post Page!

General

YPP - Your Post Page! is a platform like Tumblr, where users can register, login, upload content and see, comment and endorse others people's content, regardless of it being text, image or video based. The website is hosted at <http://ypp.theta1software.nl/>

In order to test how solid the webapp was, we tested it on Firefox and Chrome, both in computers and smartphones, and in those tests YPP worked great, but feel free to do the tests on your own. Also, for you to be able to access to the whole webapp you need to register and login, so you can see others people content and share your own content.

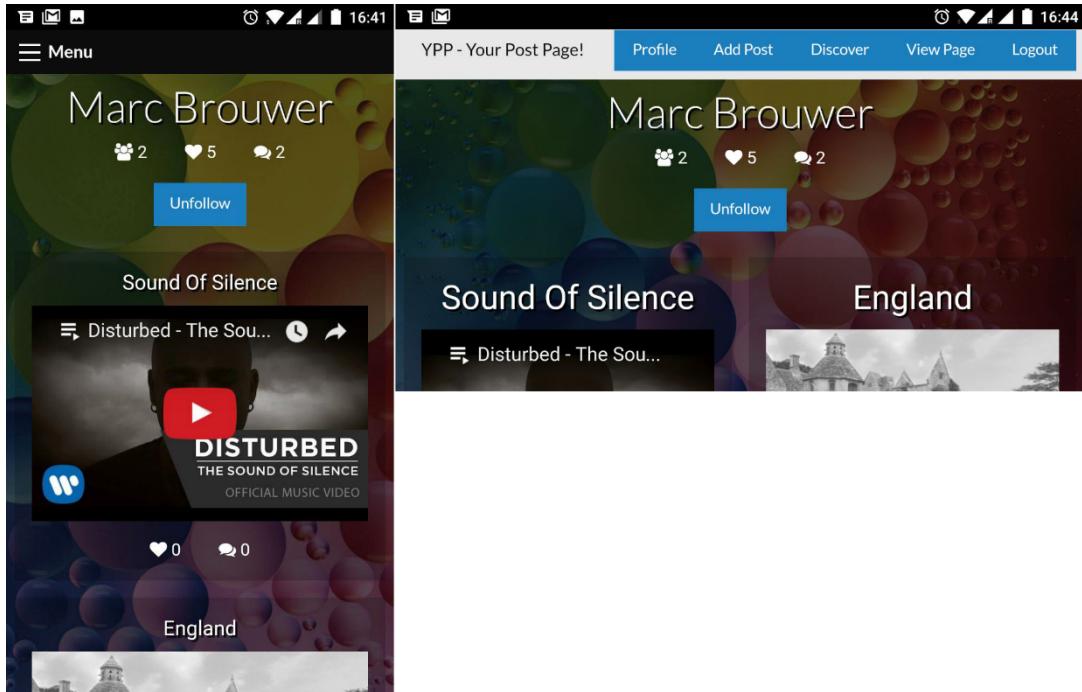
We tried to keep all the workload balanced, although for time reasons Marc has been the one who invested more hours in the most part of the areas. Jorge and Marc worked on the Database, but for the rest of the roles we all did a part of it.



Picture 1: Landing page from a laptop at full screen.

Frontend Design

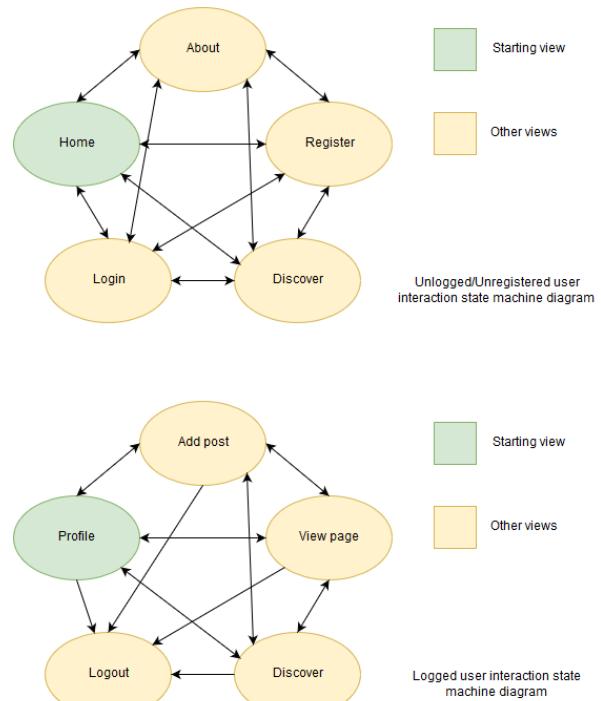
For this assignment we used Foundation as a frontend framework together with Font-Awesome, jQuery, jQueryUI, a masonry layout plugin and more additional Javascript/jQuery. The usage of this framework allowed us to create a full responsive website, easy enough to interact with from both mobile devices with small screens and full size screens.



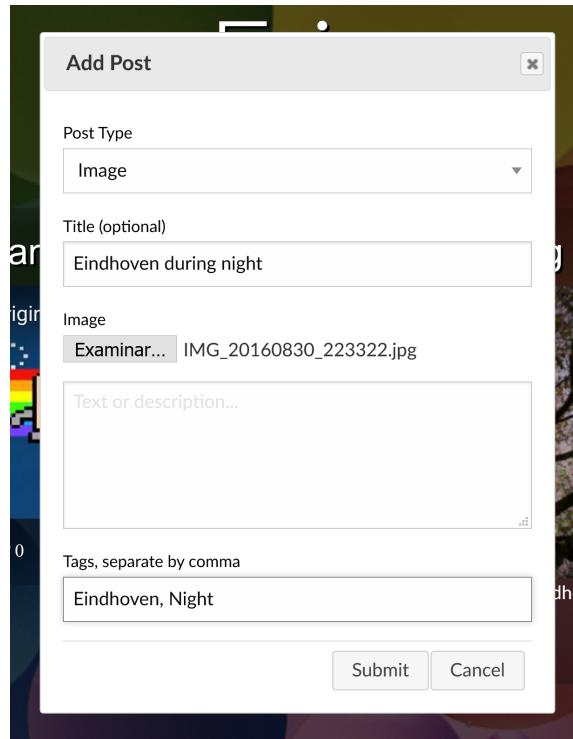
Picture 2: User's page from a smartphone, portrait and landscape positions.

The user can interact with the webapp by using the navbar at the top in order to navigate through YPP (see Picture 3).

In addition to this basic interaction, different forms will appear when necessary, like when registering or adding content (see Picture 4), but also when endorsing (pressing on the heart) or commenting content on YPP, to allow a full use and enjoy of it. For these interactions some JQuery libraries were used.



Picture 3: State machine diagrams for interaction



Picture 4: Add Post form

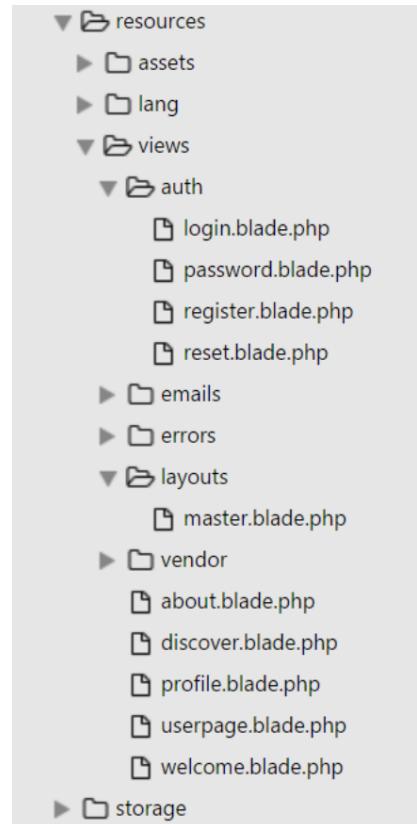
Backend Design

Laravel was the option for this assignment mainly because it is PHP which works everywhere in contrast to Django.

The base of each view/page is loaded from a so called master layout to which additional content, styles and scripts can be added. There are different kind of views, the navigation ones (discover, userpage, profile...) and the authentication ones (login, register), so they are separated in our file structure. All the routes of the project are specified in the file routes.php, so if a concrete route changes, the change will not effect views dependent on this route. The authentication views are separated from the other views as shown in picture 5.

Backend plugins for Laravel were also used. These include HTML forms, captcha and image resizing. Using these plugins made it easier to develop the backend.

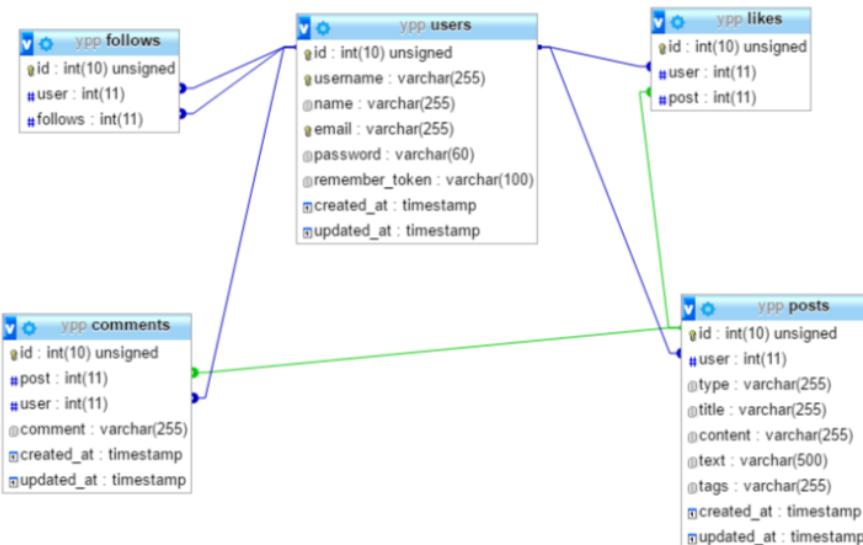
We use a MySQL Database to store all the information, that is, user credentials, posts, comments, likes and followers. It works pretty simple although some queries are complex. Views for logged in user require additional information about the posts, it should be visible to the user what he/she has liked, commented or who is followed.



Picture 5: Views structure

Databases

As said before, YPP uses a MySQL Database to store all the information. The structure of our database is very simple, yet it requires some long queries. All the date is stored in a table or another depending on its nature (user's personal information is stored in 'users', while all the posts are stored in post regardless of the user who made them, but knowing which one), and these tables are: users, likes, posts, comments and follows. You can see the structure in the picture below.



Rest API

We used Google Fonts API to be able to use its fonts, but more importantly we created our own API for the users to add posts, delete posts, edit posts, follow users, like posts, unlike posts, commenting and for discovering posts. The API routes can be found in the routes.php file. Most explain themselves. The routes, follow and like are used for both the action and undoing the action. Many routes use a limit and offset to load data as needed rather than loading all data. For example, it would not be good to load all posts when requesting the posts ordered by most likes.

Video

A video of our project can be found in the following link, and in it we show how to register, add posts, discover and follow other users and interact with their posts, and finally how to edit your data.

<https://youtu.be/tFbOTAiSN-M>

Reflection

It was the first time for the most part of us with Laravel, so we could say that we learnt its basic usage with this assignment in addition to how modern and dynamic websites are developed (not for the technique but for the teamwork and interaction). Since for the reasons of being new to Laravel and having time difficulties the work wasn't equally distributed (although it was our intention), with our current experience and knowledge we all agree on having done things differently, not only for the execution of things but for the idea itself. However, we are pleased with the result we obtained.