

Web technologies: Report

Youssef Boudiba, Thibaut Deweert, Geatan Boey, Linda de Corte

Contents

1	Introduction	3
2	the web application architecture	3
3	Implemented functionalities	4
3.1	User	4
3.1.1	The user	4
3.1.2	registering	4
3.1.3	edit user	4
3.1.4	User profile page	4
3.1.5	quizes on user profile	4
3.2	Quiz	4
3.2.1	overview page	4
3.2.2	filter	4
3.2.3	add a quiz	4
3.2.4	individual quiz	4
3.2.5	comments	4
3.2.6	search	4
4	the other requirements	4
4.1	AJAX	4
4.2	HTML 5	4
4.3	Web service	5
4.4	Google maps	5
4.5	Provide data	5

1 Introduction

You may know the game "Geoguessr", where you get a random location of google streetview and you have to guess where you are, and the closer your guess is the more points you get. Our game has the same idea of guessing a location on a map. Only in our game everyone can upload a picture with a location and an hint. Other people can then guess where the picture is made. They can guess 3 times, every time they lose 20 points and when they ask for an hint they lose half the points.

2 the web application architecture

For this application we chose to use the framework Sails.js. This is a MVC framework for node.js.

- Node.js, an execution environment for event-driven server-side and networking application
- front-end: For the front-end we chose to include Angular.js for this, because it is a very popular JavaScript framework, which extends HTML to make it more dynamic.
- Database: we chose MongoDB, this is a a very big noSQL database services. We chose to use it because it is flexible and is easy to use.

We chose Sails.js framework because it lets us write everything in the same language and takes over some tasks, for example it automatically sets up an API.

3 Implemented functionalities

3.1 User

3.1.1 The user

3.1.2 registering

3.1.3 edit user

3.1.4 User profile page

3.1.5 quizzes on user profile

3.2 Quiz

3.2.1 overview page

3.2.2 filter

3.2.3 add a quiz

3.2.4 individual quiz

3.2.5 comments

3.2.6 search

4 the other requirements

4.1 AJAX

4.2 HTML 5

We have used 4 HTML features in our application

1. GEO-location, we use this one when you are doing a quiz, it will first center the map on your location.
2. local storage, we use this to store the API secret on the users computer. So they will stay logged in.
3. form tags, we have used this when you add a new quiz, for the file upload.
- 4.

4.3 Web service

As another web service we have used Imgur to upload and request the pictures. This way we don't have to store all the pictures ourselves and we still have an easy way of accessing. So when a user adds an quiz, he or she will upload a picture, we will immediately send an PUT request it to Imgur. When they are done uploading we will get the ID and the location of the picture back.

The moment we need the picture, this can be on several locations, for example on the quiz page, on the overview page but also on the user page, we will send an GET request to Imgur to get the picture and to show it on the page.

Imgur also removes all the exif data, so we don't have to worry about removing them.

4.4 Google maps

We have used Google maps on several occasions. The first place where we use the google-maps API is when we are filtering based on location. The location the user fills in is send to Google. Google than sends back the gps locations of the location that matches closest with what the user filled in.

The next place we use the Google maps API is when a user is adding a picture. We first look if a picture has Exif location data in his or her application. If they have we make a google map with a marker in the center of the location. Otherwise we center on the VUB. The user can than move the marker. when they save we ask the Google maps API to give us the location.

And finally we use it when people guess the location, we again make a map with an marker, but this one is based on your GEO-location, if you do not want to share it we will use the location of the VUB for the start point. You can then again move the marker to chose the location. When you think you found the location, you save and we will ask the Google Maps API for the location. We will than do a calculation to see if you were close enough. If not you have 3 chances to guess, again by moving the marker.

4.5 Provide data

5 conclusion