DOCUMENTATION

The goal of this assignment was to implement the social media features in a website of our own. Besides having to create a website that has a natural flow and makes sense to what is requested, we familiarized ourselves with the developers tools used to implement these features.

Concerning the architecture, the site was thought to have a very natural flow to user, having a home page and a contacts page. Every other tab was created to implement one feature only.
The features were chosen according to what we felt were the needs of the users in a common website.

1. Basic Integration

A website was created, having in consideration its presentation and the organization of its contents.

Most of the functions were implemented using Javascript (along with the Facebook SDK) in order to provide rich functionalities to the users.

Facebook

a) Like Button

We decided to implement the “Send” button along with this button in order to try out other functionalities and layouts.

b) Feature of choice: Comment Area

We found this Facebook comment Area the most useful tool for a website, that’s why we chose it. This was implemented using the tools provided by the facebook API.

Other Services

Besides facebook, there are many other social services. There’s twitter and google for example. The reason why *+1 google button* was chosen was because one of the element of the group didn’t had a twitter account and it was easier to test. Besides, there were other Google tools and APIS being implemented in the website.
Implementing this feature (in comparison to the FB Like button) was very similar. Both features use Javascript, which made it easier to understand and implement.

Google Maps

In this section, we provide our contact information and used JavaScript Maps API to appear our studying location. It was a good basis for later.

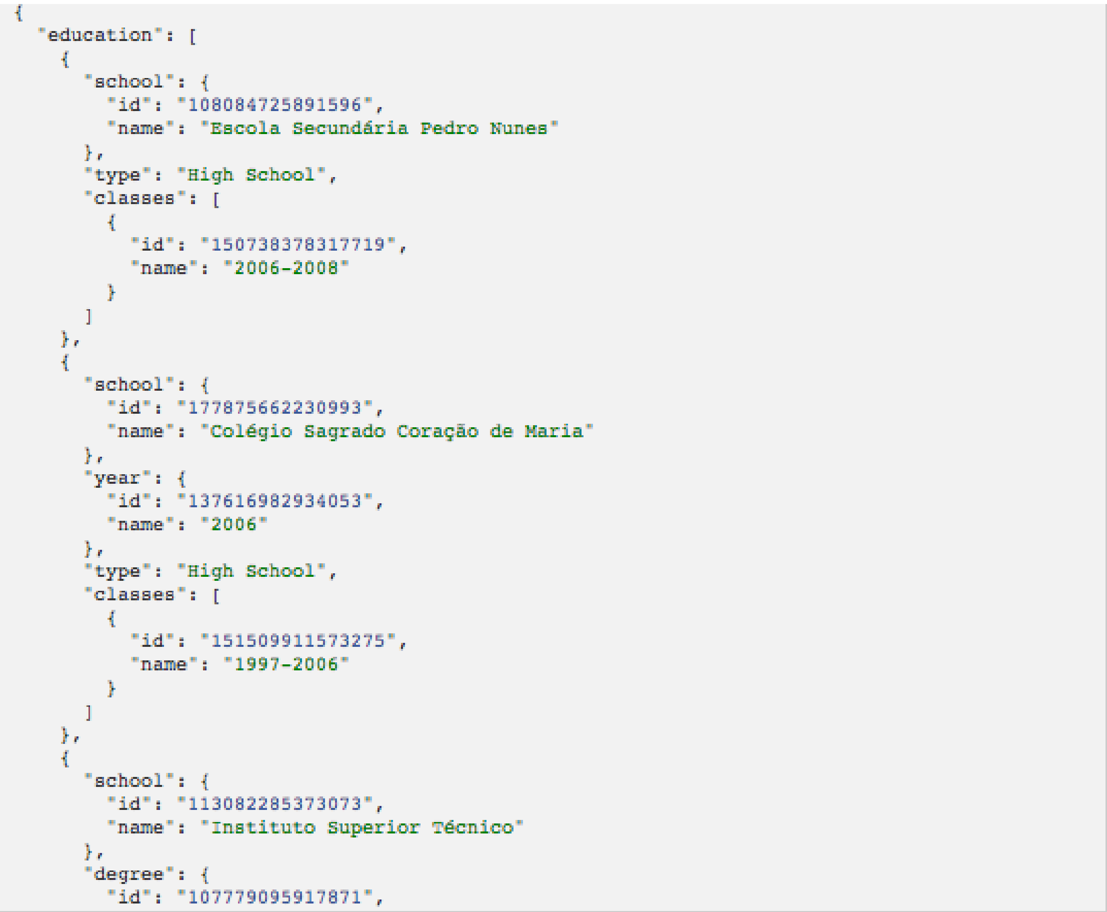
2. Deeper Integration

Authentication

In order to use any of the previous and next features (the comment area, Like button and even pushing information to and pulling from facebook) authentication in facebook is needed. So there is also a Log In button to perform this action if the user is not authenticated. Even if the user tries to perform any action (that needs facebook), he/she is redirected to the login area.

Pulling Information

Once the user is authenticated, we are able to get his information, like the name, username, photography, birthday, email, facebook link and we decided to get the user's academic background, which is something that took a little bit more time since a person can have more than one school and the way to pull the information about the school name, type and classes is not trivial due to how the information are stored in the array (see the figure below, show how the array is organized).



```
{
  "education": [
    {
      "school": {
        "id": "108084725891596",
        "name": "Escola Secundária Pedro Nunes"
      },
      "type": "High School",
      "classes": [
        {
          "id": "150738378317719",
          "name": "2006-2008"
        }
      ]
    },
    {
      "school": {
        "id": "177875662230993",
        "name": "Colégio Sagrado Coração de Maria"
      },
      "year": {
        "id": "137616982934053",
        "name": "2006"
      },
      "type": "High School",
      "classes": [
        {
          "id": "151509911573275",
          "name": "1997-2006"
        }
      ]
    },
    {
      "school": {
        "id": "113082285373073",
        "name": "Instituto Superior Técnico"
      },
      "degree": {
        "id": "107779095917871",
```

We used the Graph API to pull the information with the auxiliary tool Graph API Explorer to test. (https://developers.facebook.com/tools/explorer)

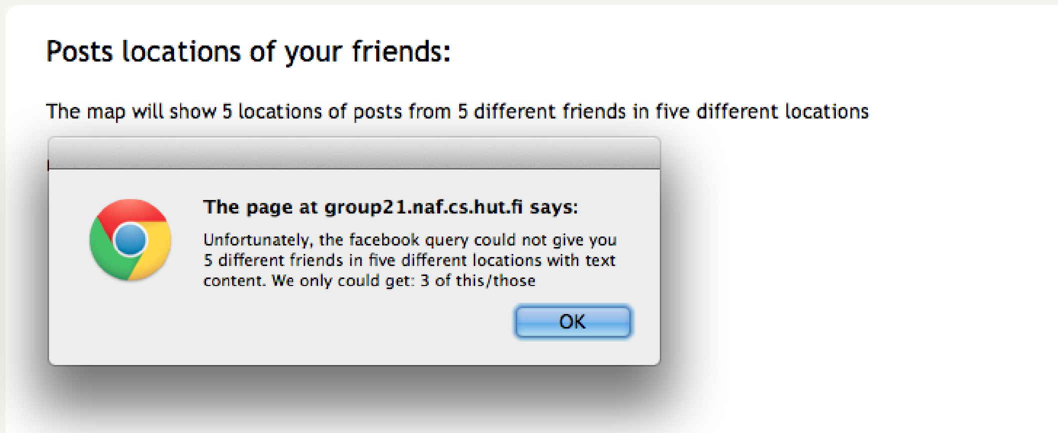
Pushing Information

In the same way the user can see their information, it is also possible to upload either one or more pictures with the respective description into his/her facebook account. This will create an album in the user's facebook page, containing the uploaded picture/s.
There is also the possibility of the user posting a comment on his wall using our application.

FB Canvas and Google Maps

The FB Canvas feature was implemented by hosting the webpage inside a facebook app. So if the user clicks on the "Canvas" tab, it will redirect to the facebook app where the implemented page can be seen and used there.

The google maps feature was the most difficult to implement, as it was implemented using not only the google API to fetch the map and its information, it also required the use of FQL. With this tool we were able to perform many queries at a time (in a single call), optimising the information's search time and allowing us to show up to five different friends on the map. Also, we could get informations (longitude and latitude of a friend's post) that were not as easy to get with the Graph API.
The time spent on this feature was high, due to all the restrictions and cover all the cases where the application could fail . Some problems we faced it was enters (new lines), single and double quotes of the users posts and after do the bridge between php(FQL query) and JavaScript (Google Maps). FQL query despite of going deeper face to Graph Api, has some limitations (e.g it is not possible to join two tables on the same query).
The google maps always shows 5 different friends in five different locations with text content except if the FQL doesn't give enough data to satisfy the requisites then it will alert the user that it only could give x friends . (As an example, see the figure below).



Posts locations of your friends:

The map will show 5 locations of posts from 5 different friends in five different locations

The page at group21.naf.cs.hut.fi says:

Unfortunately, the facebook query could not give you 5 different friends in five different locations with text content. We only could get: 3 of this/those

3. Additional Exercises

We implemented the FQL feature to optimize and go more deeper on the tab friends making possible to see who posted and the content of the post linked with location on google maps.

In addition we used the Youtube API to load a video (home tab), that we create on the YouTube tab. The YouTube tab enables to create and upload the video to youtube, saving time to the user (who can do it all at once).

Limitations

- Whenever someone adds a photo (in tab Post), it redirects to a page that contains the created token instead of redirecting to the website's page.

- Https problem already explained on the README . (**Deprecated** - changed after demo. https worked after server problems were solved)

Working hours per person:

- Catarina Moura - 37 hours

- Inês Castelo - 37 hours