Social Media and Mashups

In this assignment we ask you to build two web applications (or one application that covers the both purposes). One will be a static webpage, while the other will contain some server-side code. Simple prototypes will suffice; we do not require fancy graphics or a commercially feasible idea. Please focus on the technical aspects rather than sociological or business perspectives.

Suggested Workload and Grading

The target workload for this assignment is 25 hours per student (working in pairs). In addition to the points mentioned with each exercise, **the course staff may give at most 6 extra points** about an extraordinarily good application that has a great idea, spectacular presentation and that uses the techniques in a creative way. The lower limits for each grade are as follows:

1: 8
2: 11
3: 14
4: 17
5: 20

Deliverables

Please add a new directory called *assignment_2* in the SVN repository. Then put the following items into *assignment_2*. Please also follow the general guidelines.

- application demo: put the URL into README
- source code
    - remember to specify the source of all borrowed code clips
- README
    - brief installation instructions
    - if any: tell about library dependencies and the required special configurations
- documentation
    - description of work completed
    - a description of application architecture and a short reasoning for it
        - if applicable: discuss also a bit about networking architecture
    - a demonstrative use case
        - e.g. some kind of sequence of typical user action
    - if any: further ideas, comments
    - spent working hours per person

Environment

You are provided with a web server where you can run programs in many languages and on several platforms. For further information, please have a look at the instructions for the server usage at Noppa page of the course. Since the exercises can be done with a minimal amount of server-side programming, selection of the language is up to you. To minimize the overhead from web programming, we recommend that you pick a framework that you are already familiar with. If you are not familiar with any of the technologies that we provide to you, you might want to use PHP. **Remember that this course is about network application frameworks, not web programming**.

If you wish to try out Amazon EC2, please contact us. We may be able to provide you some credits for Amazon so that you don't need to pay for it by yourself. However, the effort to learn EC2 is not included in the workload estimate of the assignment. Instead, you acknowledge that this might cause some extra work for you in exchange for learning a bit more.

1. Basic Integration (6p, Mandatory)

(Ref: **http://developers.facebook.com/docs/plugins/**)

Create a simple static website and extend it with dynamic content from social media services. In each of the steps in this exercise you should:

- Test out different types or layouts of a plugin (if applicable),
- Demonstrate a feature on your website,

- Report your experience shortly in the documentation.

Facebook

1. **"Like" Buttons**
   Social media services act also as advertising channels for commercial Internet services such as newspapers. Facebook's Like button is probably the simplest way to advertise a service through the users' activities in their own communities. It is no wonder that we see those buttons so often on popular websites. Due to its easiness, adding a button to your own website is also a good way to get started with Facebook's developer material.

2. **A Feature of Your Choice**
   Aside from Like buttons, there are many other plugins. For instance, you could have a blog in a static website and allow people to comment on your posts with Comments plugin. Please look at some examples provided by Facebook to get a better idea.

Other Services

Use another social media service (e.g. Twitter) to do a similar thing that you did with Facebook. Compare the basic integration techniques of the service to those used in Facebook.

Google Maps

Add a "contact information" section to your webpage and include a map using Google Maps to help potential visitors see your location. Use JavaScript Maps API to provide the user with a scrollable map.

2. Deeper Integration (12p, Mandatory)

Build a dynamic web application that uses the information received from social media services.

Authentication

Implement authentication for your webapp using Facebook as the identity provider. There are some ready-to-use code clips available for this purpose, which you are allowed to use, provided that you demonstrate your understanding of the technology in your submission in a way or another.

Pulling Information

Add a feature that makes use of user data that is fetched from Facebook. Rather than just greeting the user with his/her name, please try to use your creativity to pull out something a bit more interesting. This exercise is supposed to be completed with Graph API, but you may find other ways as well (for instance, please look at the exercise 3).

Pushing Information

Add a feature that pushes some information to Facebook. For instance, you could make a photo uploader. You might want to use *Dialogs*, but please also concentrate on exploiting the features of the Graph API.

FB Canvas and Google Maps

Present some geographical information about the user and his/her friends on a map using Google Maps JavaScript API. One good idea would be to show home places. Put your application "inside" Facebook using Facebook's Canvas feature. Please note that using FBML is already considered a deprecated way of doing things, so you may use the newer iframe solution.

3. Additional Exercises (6p, Optional)

You may get six points from any of the following exercises. A solution at the level of proof of concept (PoC) is accepted, but gives you only half of the points. For full points, you will need to do either **1)** two exercises as PoC, or **2)** one well-done exercise. We evaluate at most two exercises per student group.

FQL

There are many situations, where Graph API is not efficient enough. Facebook offers another way to run queries on their database, namely FQL. For instance, you could use FQL to optimize the performance in

FB Canvas exercise.

Google APIs

Google provides developers with dozens of different APIs. Pick **one** of the following alternatives and provide an application that uses the API.

- Chart Tools
- Calendar
- YouTube
- Analytics
- Blogger Data

Your Own Topic?

In addition to the previously mentioned topics, you might want to try out many other things. If you are up to something special, we would be glad to hear from you! Before you start implementing your idea, please write a short non-technical description of your idea (max. 100 words), your first thought of technical implementation (no more than half a page) and send it to us by email for approval. Your description does not affect on the grade, but we reserve the right to reject the idea. We will answer to to you within 2 business days, after which you can start working on the topic.