

# ONE SOCIAL

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment of the

Requirements for the

Degree Master of Computer Science

May 2010

Sowmya Sampath

## ABSTRACT

The growth of social network websites Facebook and Twitter has been exponential. Hence, leading companies like Yahoo and Microsoft have added these two social networking websites as modules (applications) in their respective home pages. However, in their home pages, they exist as two different applications and do not interact with each other. In other words, the Facebook streams and Twitter tweets are shown in two different views and users are required to update both these modules separately. There is currently no application that shows both the Facebook streams and Twitter tweets in a single view. Users find it difficult to manage and maintain both these social networks together. My goal is to build a social website, One Social, where users can view their Facebook streams and Twitter tweets in one single view and thereby easily organize and maintain their updates.

In One Social, users can enter a status message and can share it with both their Facebook and Twitter connections in just one click. Moreover, all the Facebook and Twitter comments for a status will be grouped (or threaded) together when I display them in One Social to improve user experience. They will also be able to set various preferences for status sharing. Users can create a single One Social ID and link both their Facebook and Twitter IDs to this One Social ID. Thus, logging onto One Social will log on the user to both these social networks. I will be using the REST API's provided by both Facebook and Twitter to view and update stream data.

# Table of Contents

1. Introduction.....	4
2. Deliverables.....	6
2.1 Curl and MultiCurl .....	6
2.1.1 Steps followed to cURL.....	6
2.1.2 Multicurl.....	7
2.2 Facebook and Twitter Authentication.....	8
2.2.1 Facebook authentication .....	8
2.2.2 Twitter authentication.....	10
2.3 Get and Update Status in Twitter.....	11
2.3.1 Get Tweets from Twitter.....	12
2.3.2 Get Followers .....	12
2.3.3 Follow Friends.....	13
2.3.4 UnFollow Friend .....	13
2.3.5 Update Status.....	14
2.4 Get and Update Status in Facebook .....	14
2.4.1 Get Stream from Facebook.....	15
2.4.2 Get Friends from Facebook.....	16
2.4.3 Set Status .....	16
3. Conclusion and Future work .....	17
APPENDIX .....	18

# 1. Introduction

Facebook and Twitter have become a part of our lives. Maintaining friendships and contacts over phone and email has become cumbersome when online social networking through blogs and status messages is the easy way out. Building social networking websites has now become a billion dollar industry, and slowly numerous companies have started building their own social networking sites. Facebook, Twitter, MySpace, LinkedIn and Orkut are some of the popular ones. People who are unaware of the sites or who don't possess an account with at least one of these are feeling left out. As a member of few of the above mentioned websites, a user might have trouble maintaining all of them at the same time.

A typical user of any one of the social networking websites would create a profile, update his status or check the statuses of his friends at least 10 to 15 times a day. This also includes adding comments to some of the statuses, adding a friend, accepting a friend request, or removing a friend from his list. Features that are provided by the social network sites increase often, and the user needs to learn adapt and maintain the networks too. The amount of work a user does for one such website is quite time consuming, and the work doubles when he/she decides to join another networking site.

Facebook and Twitter have been the most popular social networking websites of all time. Creating accounts, maintaining statuses and adding comments on these two websites might increase the work of the user. In my project I am designing a scalable website which would integrate Facebook and Twitter accounts of a user, so that user can comfortably view and update his status and comments from a single place.

I chose Facebook and Twitter not only because they were popular but mainly because they were both open source. The APIs/web-services provided free by these two websites make it possible for this task. Once the other popular websites start providing free APIs, they too can be incorporated and integrated in my website with very little work.

The main attractive feature that can be seen in my One Social website is that it not only pulls data from Facebook and Twitter but it also updates both of them simultaneously

with a single click of a button. Many users find it common to update their statuses to both Facebook and Twitter. This cool feature of One Social enables the user to enter their status in a single text box present in my website and on the click of a button it gets updated to both Facebook and Twitter. Next, when the status message receives a comment on it, either in Facebook or in Twitter, they get grouped and shown at a single place. The friends from Facebook and Twitter are differentiated with a Facebook/ Twitter icon near them. The following would be a sample prototype design of the final website.

My image

I am going to hawaii for the summer !!!

User image **Friend from college :**  
Hooray! have a great time :)  
  
Comment

User image **One of my cousins :**  
Hey gr8! Bring sme soveniers :P  
  
Comment

User image **Old school friend :**  
OOps! fancyyyyyy :D

User image **My Mom :**  
Hey ! take care :)  
  
Tweet

For this semester I have implemented the Facebook and Twitter updating statuses independently from One Social.

## 2. Deliverables

### Curl and MultiCurl

Curl and MultiCurl are the most important and inevitable components for this project. PHP has provided a wide library for php documentation at php.net, which provides the APIs and the function calls needed to perform curl and multicurl.

Curl is one of the most powerful PHP extensions. It stands for Client URL, and allows you to communicate with other servers using a wide range of protocols. I used curl to fetch a RSS feed from cnet.com. Curl does not come with the PHP installation. It needs to be installed separately.

Curl will be used in my project in various places where I need to get data from Facebook/Twitter. So as a first deliverable I am using curl to experiment pulling data from cnet.com.

#### 2.1.1 Steps followed to cURL

The following are the steps to be followed to curl data from a url.

- 1) Get the curl handle by initiating it.

```
$ch = curl_init();
```

- 2) Retrieve the data from the url

```
curl_setopt($ch, CURLOPT_URL, "http://news.cnet.com/8300-17938_105-1.xml?tag=rtcol;about");
```

- 3) Return the data back as string in curl\_exec

```
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
```

- 4) Check out the other\_setopt options (like setting timeout in milliseconds).

```
curl_setopt($ch, CURLOPT_TIMEOUT_MS, 400);
```

#### 5) Execute the curl

```
$response = curl_exec($ch);
```

Once the data is retrieved from the url, I can process it based on the data being retrieved. It would be of 3 kinds

- 1) String
- 2) XML
- 3) JSON (Javascript Object Notation).

In this case, I wanted to process it as an XML, so I extracted the data iterating the XML content and retrieving the values held in the tags. Curl returns lots of specific error codes that could help us identify the exact issue, in case of a failure. Also, setting the right timeout for a curl call is very essential. Setting a very high timeout value, could at times, bring down our own site, when the publishers site is down. Setting a very low timeout value would result in a bad user experience. Hence, it is essential to benchmark an API before setting a timeout value. `CURLOPT_VERBOSE` is another useful curl option when trying to debug HTTP issues. It dumps the complete request including the headers and the corresponding response data and headers.

### 2.1.2 Multicurl

Multicurl allows processing of multiple cURL handles in parallel. Using multi curl, I can fetch the contents of various urls in parallel. PHP does not support multi threading and multi curl is a very nifty way of fetching things in parallel to improve performance. Without multi curl, I will only be fetching data from various urls sequentially. When building a site like OneSocial, where data is pulled at various times from multiple Facebook/Twitter APIs, it is impertive to use multi curl for performance optimization. For a sample multicurl PHP experiment, I have still continued with cnet.com and extracted multiple RSS feeds and processed them as XML.

The steps involved for multicurl are:

- 1) Get the curls from cnet.com
- 2) Create a multi-handle and add all the curl-handles to it.

```
curl_multi_add_handle($mh, $ch[$i]);
```

- 3) Execute the multicurl

```
curl_multi_exec($mh, $running);
```

Multi curl offers all the functionality offered by curl with respect to error codes and verbose mode making it easy for debug issues.

## 2.2 Facebook and Twitter Authentication

Facebook and Twitter have got their APIs specified along with their parameters in these pages.

Facebook wiki - [http://wiki.developers.facebook.com/index.php/Main\\_Page](http://wiki.developers.facebook.com/index.php/Main_Page)

Twitter wiki - <http://apiwiki.twitter.com/Twitter-API-Documentation>

This deliverable is to show how the authentication process has to be set up in One Social. This is because people signing in to One Social are giving their credentials to a website, which might misuse them. This can be prevented by using https and various other measures to protect the user credentials. First, I provide a means for the user to get into Facebook/Twitter from One Social. For this, I have looked into the authentication protocols of the websites.

### 2.2.1 Facebook authentication

The following steps are done for the authentication/authorization process of Facebook.



1. Call Auth.createToken to get a token from FB
2. Generate a FB login url with which users enter their uname/pwd
3. FB returns a session key once the user successfully both authenticates and authorizes One Social. I will have to specify a call back url to which FB returns the session key via POST data.
4. Use this session key to make subsequent FB API calls.

A sample Facebook API call

```
function getAuthToken() {
    $params = array('api_key' => API_KEY,
        'v' => '1.0',
        'format' => 'json',
        'method' => 'facebook.auth.createToken');
    $sig = generateFBSignature($params, SECRET);

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "http://api.facebook.com/restserver.php");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS,
"method=facebook.auth.createToken&v=1.0&api_key=" . API_KEY .
"&format=json&sig=$sig");
    curl_setopt($ch, CURLOPT_POST, 1);

    $out = curl_exec($ch);
    $auth = json_decode($out);
    curl_close($ch);

    return $auth;
}
```

Facebook provides the following two application configurations for application developers:

- 1) Post-Authorize Callback URL: After a user authorizes your application, Facebook sends a number of POST parameters to your Post-Authorize Callback URL in the form of a POST request.
- 2) Post-authorize redirect URL: When a user authorizes your application, that user is redirected to this URL.

I have had various issues in the past where post authorize redirect URL did not redirect the user back to OneSocial after successful authorization. Part of the problem is that this redirect URL is only called when a user authorizes OneSocial for the “first time” and not during the subsequent times. Also, this URL must not be more than 100 characters.

## 2.2.2 Twitter authentication

The following steps are done for the authentication/authorization process of Twitter.

1. Obtain the unauthorized request token.
2. Twitter grants the Request token
3. Resend it with user credentials
4. Obtain the Access token
5. Use this for further communication

A sample Twitter API call

```
function getRequestToken() {  
    $params = array('oauth_consumer_key' => API_KEY,  
                  'oauth_signature_method' => 'HMAC-SHA1',  
                  'oauth_nonce' => md5(microtime() . mt_rand()),  
                  'oauth_timestamp' => time());
```

```
$sigParams = getSignedHeader("http://twitter.com/oauth/request_token",  
$params);
```

```
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, "http://twitter.com/oauth/request_token");  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
curl_setopt($ch, CURLOPT_HTTPHEADER, array($sigParams));
```

```
$out = curl_exec($ch);
```

```
if(curl_errno($ch)) {  
    print 'Curl error: ' . curl_error($ch); exit;  
}
```

```
parse_str($out, $auth);  
return $auth;  
}
```

Twitter's three legged OAuth process is much simpler compared to Facebook and I were able to get it to work in the very first attempt.

### 2.3 Get and Update Status in Twitter

After achieving authentication in Twitter, next thing to do is to get data and update data into Twitter. First I start with Twitter and get tweets, get friends list (people who I follow in Twitter), follow and unfollow friends and atlast update status in Twitter.

### 2.3.1 Get Tweets from Twitter

In Twitter, tweets are daily status or events, which the user updates and people who follows them can view, and update them. I display the most recent statuses, including retweets, posted by the authenticating user and that user's friends.

API Usage:

- a) URL: [http://api.twitter.com/1/statuses/home\\_timeline.json](http://api.twitter.com/1/statuses/home_timeline.json)
- b) Request Parameters:
  - a. Auth token – This is the user's secret that uniquely identifies a user
- c) Request Method: GET
- d) Response data: All the statuses of the authenticated user and for each of the status, their associated retweets and all the user information who retweeted.

Getting tweets from Twitter based on the above steps may look straightforward but the parameters required for the API requires more understanding of the Twitter architecture.

### 2.3.2 Get Followers

Following someone on Twitter means you are subscribing to his or her Tweets, and their updates will appear in your personal timeline on your Twitter homepage. The following API is used to get the list of followers for a given user.

API Usage:

- a) URL: <http://api.twitter.com/1/statuses/friends.json>
- b) Request Parameters:
  - a. Auth token – This is the user's secret that uniquely identifies a user
- c) Request Method: GET

- d) Response data: Returns a user's friends, each with current status inline. They are ordered by the order in which the user followed them, most recently followed first.

### 2.3.3 Follow Friends

If you follow a user, you'll get their updates on your homepage when you log in. When you follow someone, they will then have permission to send you private tweets, called direct messages too. The following API is used to follow someone in twitter.

API Usage:

- a) URL: [http://api.twitter.com/1/friendships/create/\\$friendID.json](http://api.twitter.com/1/friendships/create/$friendID.json)
- b) Request Parameters:
  - a. Auth token – This is the user's secret that uniquely identifies a user
  - b. Friend ID – ID of the friend to follow
- c) Request Method: POST
- d) Response data: Returns the befriended user in json format when successful. Returns a string describing the failure condition when unsuccessful.

### 2.3.4 UnFollow Friend

Sometimes the list of friends I follow would become too big to have a maintainable status updates. Removing someone from your following list means that you will no longer receive his or her updates.

API Usage:

- a) URL: [http://api.twitter.com/1/friendships/destroy/\\$friendID.json](http://api.twitter.com/1/friendships/destroy/$friendID.json)
- b) Request Parameters:
  - a. Auth token – This is the user's secret that uniquely identifies a user
  - b. Friend ID – ID of the friend to unfollow
- c) Request Method: POST

- d) Response data: Returns the unfollowed user in json format when successful. Returns a string describing the failure condition when unsuccessful.

### 2.3.5 Update Status

Sometimes the list of friends I follow would become too big to have a maintainable status updates. Removing someone from your following list means that you will no longer receive his or her updates.

API Usage:

- a) URL: <http://api.twitter.com/1/statuses/update.json>
- b) Request Parameters:
  - a. Auth token – This is the user’s secret that uniquely identifies a user
  - b. Status – The status message to be posted
- c) Request Method: POST
- d) Response data: Returns HTTP status 403 when the status message is more than 140 characters or when the current status is the same text as the most recently posted status. On success, all the meta information of the status message and the authenticated user is returned.

## 2.4 Get and Update Status in Facebook

Similar to Twitter, after achieving authentication in Facebook, next thing to do is to get data and update data into Facebook. I will look at how to get the Facebook stream of a user, get a user’s friends list and at-last update status in Facebook.

## 2.4.1 Get Stream from Facebook

API Usage: (Get Stream)

- a) URL: <http://api.facebook.com/restserver.php>
- b) Request Parameters:
  - a. Session Key – This is the user’s secret that uniquely identifies a user
  - b. Method - facebook.fql.query
  - c. FQL Query – The fql query that returns a user’s stream data. I will need to pass the user’s profile id in the fql query. (Ex: `SELECT post_id, actor_id, target_id, message, comments FROM stream WHERE source_id = 100001048604143 and is_hidden = 0`)
- c) Request Method: POST
- d) Response data: All the statuses of the authenticated user and for each of the status, their associated comments, and the user ids who made these comments.

API Usage: (Get User Data)

- a) URL: <http://api.facebook.com/restserver.php>
- b) Request Parameters:
  - a. Session Key – This is the user’s secret that uniquely identifies a user
  - b. Method - facebook.fql.query
  - c. FQL Query – The fql query that returns the Meta information of a list of users. I will need to pass the list of user profile ids in the fql query. (Ex: `SELECT uid, name, pic_small FROM user WHERE uid IN ($actorIDs)`)
- c) Request Method: POST
- d) Response data: The name and profile picture of all the users.

## 2.4.2 Get Friends from Facebook

API Usage: (Get Friends IDs)

- a) URL: <http://api.facebook.com/restserver.php>
- b) Request Parameters:
  - a. Session Key – This is the user’s secret that uniquely identifies a user
  - b. Method - facebook.friends.get
- c) Request Method: POST
- d) Response data: All the IDs of the people who are friends with the current authenticated user.

API Usage: (Get Friends Names)

- a) URL: <http://api.facebook.com/restserver.php>
- b) Request Parameters:
  - a. Session Key – This is the user’s secret that uniquely identifies a user
  - b. Method - facebook.fql.query. I will have to pass the friends ids whose names have to be retrieved. (Ex: select name from user where uid IN (1121210, 343430))
- c) Request Method: POST
- d) Response data: All the Names of the people whose ids are passed to the fql query.

## 2.4.3 Set Status

API Usage:

- a) URL: <http://api.facebook.com/restserver.php>
- b) Request Parameters:
  - a. Session Key – This is the user’s secret that uniquely identifies a user
  - b. Method - facebook.status.set
  - c. Status – The status message to be posted to Facebook



- c) Request Method: POST
- d) Response data: 1 on success, 0 on failure or an error code.

### 3. Conclusion and Future work

One social has several more steps for completion. To start, One Social needs to be matched with a stable front end. The integration of Facebook and Twitter will be the next task, which would hold many challenges: threading the statuses, updating statuses in one place and thereby updating it in both Facebook and Twitter, setting a stable database to link the One Social ID with Facebook/Twitter id, and handling security for user credentials.

I need to host the website to bring in user load. Hosting the website can either be through the port-forwarding technique from my router or paying companies like Yahoo for a domain. Performance operations like memcached and/or squid are implemented when the user load becomes considerable.

Testing the One Social website is also an important phase of my project. Testing and bug fixing enables me to obtain a more stable and clean implementation. Documentation of the code will be done to provide proper understanding of the code and room for future enhancements. All the above mentioned phases would be implemented by December 2010.

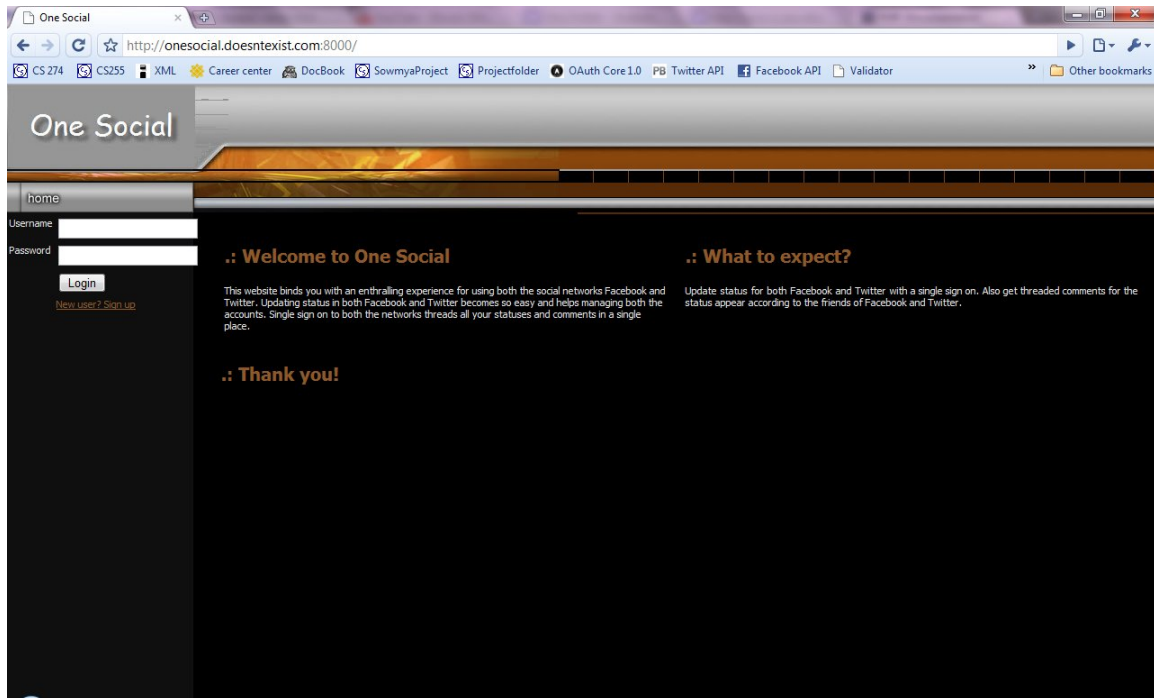
### 4. References

[1] [http://wiki.developers.facebook.com/index.php/Main\\_Page](http://wiki.developers.facebook.com/index.php/Main_Page)

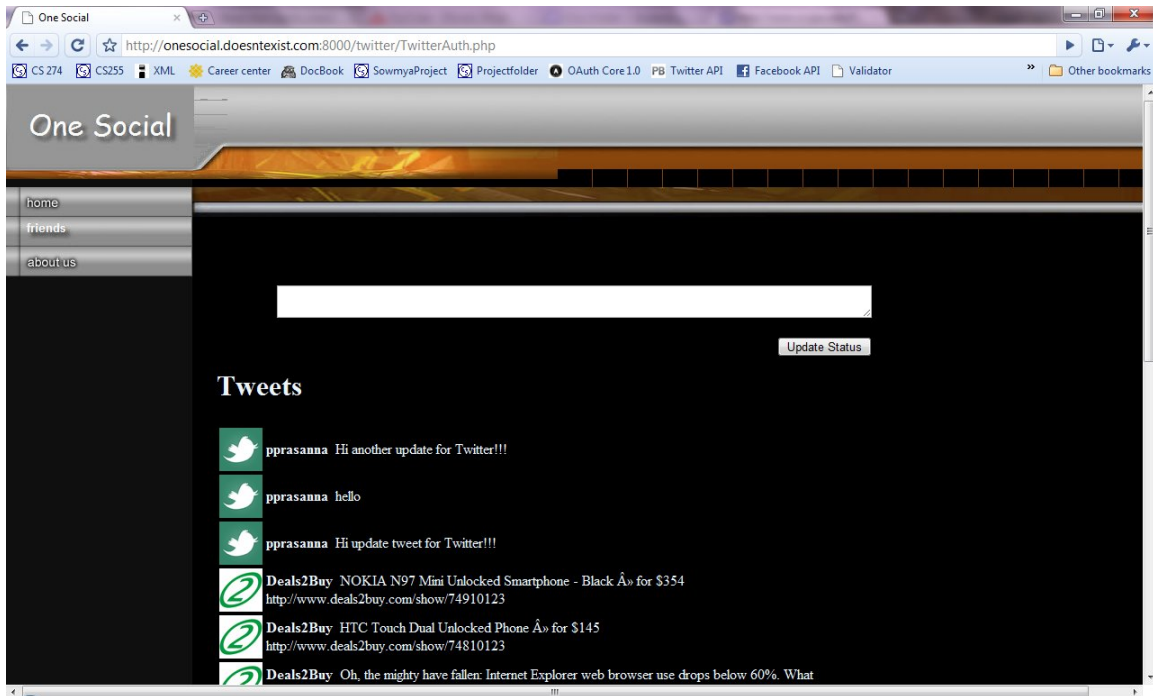
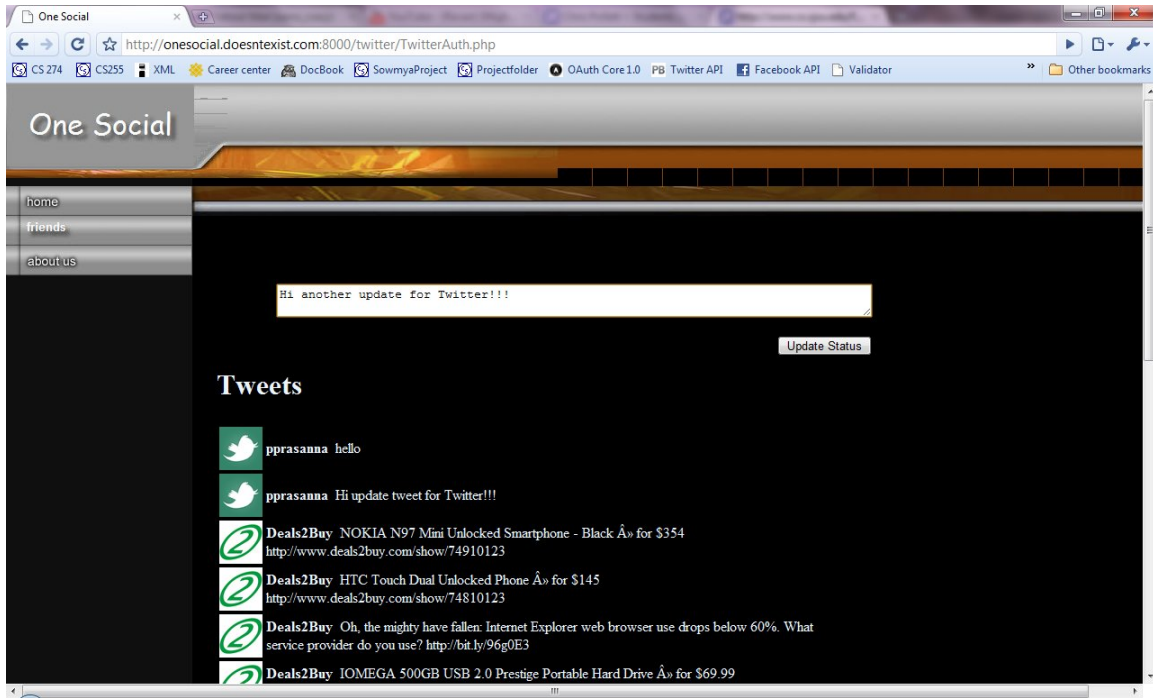
[2] <http://apiwiki.twitter.com/Twitter-API-Documentation>

[3] <http://www.php.net/docs.php>

# APPENDIX



# Twitter Update



# Facebook Status Update

