

WEBD 236 Lab 1

If you use an external source (i.e. a web-page, the required textbook, or an additional book) to help you answer the questions, then be sure to cite that source. You should probably always be citing a source.

Problem

URLs are way too long, especially for sharing with other people via Twitter or Facebook. To solve this problem, URL shortening web sites such as bit.ly have been created. Your assignment is to write a simple URL “shortener” web application called “SmallURL.”

SmallURL will serve two main tasks: URL shortening and click tracking. When a user submits a URL for shortening, you will generate a new “shorter” (it may actually be longer) URL that references a database of URLs mapped to IDs. For example, when someone shortens <http://www.franklin.edu/>, you may generate a new address such as <http://localhost/smallurl/getUrl.php?id=1>. What this basically means is that ID #1 in the database will contain the URL that was shortened as well as a count of times that the URL has been clicked. When the shorter URL is clicked, you will need to increment the click count in the database. An attempt to access an invalid ID should just result in a message that the URL wasn’t found.

Here are some example screen shots:

SmallURL Site

Hello. Please submit a URL that you would like to shorten.

URL:

Top 5 clicked URLs

URL	Clicks
http://www.franklin.edu/	2
http://cnn.com/	0

[Home](#)

Note that in the above, the user is shortening the link to xkcd.com. Also notice that the top 5 clicked URLs are displayed on the home screen in a tabular format. Clicking on these links should increment the click count for that URL because while the original text is displayed, the actual link is the shortened one.

SmallURL

<http://xkcd.com/> shortened to <http://localhost/smallurl/getUrl.php?id=3>

[Home](#)

After shortening the link to xkcd.com, the above page is displayed. This shows the link that the user may employ for click tracking.

No such url found.

[Home](#)

If the user changes the ID number in the URL manually, and tries to fish for a link that doesn't exist, then the above message should be displayed.

One caveat: shortening a link that already exists in the database should *not* generate a new tracking link. Instead, the old link should be given back. In other words, if we try to shorten the link to xkcd.com a second time, we'll still get back ID number 3, not a new ID.

Helpful Hints

- You'll only need one database table called "urls." This table should have three columns: the ID number, the original URL, and the click count.
- You should use SQLite (not MySQL) for this project. A very simple "db.inc" include file will help get you started:

```
<?php
$db = new PDO('sqlite:smallurl.sqlite');
$dbCreated = false;

function checkDb() {
    global $db;
    global $dbCreated;

    if (!$dbCreated) {
        $db -> beginTransaction();
        $q = $db -> query("SELECT name FROM sqlite_master WHERE
type='table' AND name = 'urls'");
        if ($q -> fetch() === false) {
            $db -> exec("CREATE TABLE urls (id INTEGER PRIMARY
KEY, url CHAR(255), count INTEGER)");
        }
        $db -> commit();
        $dbCreated = true;
    }
}

checkDb();
?>
```

The above code defines a function called `checkDb` that will look for the existence of a table called "urls" in the file called "smallurl.sqlite." If that table doesn't exist, then it will create it in the current directory. This file is also a fine location to define other functions that access the database (such as `findUrlById`, or `addUrl`).

- You should likely have at least 4 PHP files:
 - `db.inc` (mentioned previously)
 - `index.php`: for the main page with the text field and 5 top links.
 - `addUrl.php`: the target action for the POST from `index.php`.
 - `getUrl.php`: the action that redirects to the target page.

You may have other include files (like headers and footers or a file of utility functions) and style files as needed.

- The actual redirection to another location is handled through the PHP function called “header.” If no output is sent to the browser (i.e. not even a doctype), then you can use the header to redirect as in the example below:

```
header('Location: http://xkcd.com/');  
exit();
```

The call to “exit” terminates the current script. The browser will then redirect to the given URL.

Submission instructions

Create a ZIP file of your entire project contents as it is found under your C:\XAMPP\htdocs\smallurl\ directory. This ZIP file should contain your database, your PHP scripts, style files, etc. Follow proper coding conventions (indentation, commenting, etc.) that you would have learned in previous courses.

Submit the ZIP file to the dropbox for this assignment in the course.