Construction of an Indiana Water Monitoring Inventory Using the Google Maps API

Jae Sung Kim
Purdue University, kim326@purdue.edu

Laura Esman
Purdue University

Jane Frankenberger
Purdue University

Follow this and additional works at: http://docs.lib.purdue.edu/gisday

http://docs.lib.purdue.edu/gisday/10

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
The Construction of Indiana Water Monitoring Inventory using Google Maps API

Jae Sung Kim\textsuperscript{1}, Laura Esman\textsuperscript{2}, Jane Frankenberger\textsuperscript{2}

\textsuperscript{1} Civil Engineering, Purdue University
\textsuperscript{2} Agricultural and Biological Engineering, Purdue University

\textsuperscript{1} jaesung@purdue.edu
Need for Inventory

- Monitoring conducted at many locations within Indiana’s waters by a variety of government agencies and organizations
- The information and location may be difficult to find

- Displays information and location
- Determine what data have been collected
- Lets user contact the data holders or their web site for more information
- Enables data holder to upload water monitoring information
Indiana Water Monitoring Inventory

A central hub for water monitoring locations of Indiana streams, lakes, and groundwater.
GoogleMaps API (Application Programming Interface)

- What is it?
  GoogleMaps utilities programmed in Java Script, which can be embedded in any webpage

- Why GoogleMaps?
  - Supports variety of types of imagery such as map, satellite image, and terrain map.
  - Any customized application can be integrated conveniently
  - User Friendly
How inventory works

Load

MySQL Database

PHP Variables

Javascript variables (GoogleMaps API variables)

Display
What is Javascript?

- designed to add interactivity to HTML pages
- usually embedded directly into HTML pages
- an interpreted language
- Everyone can use JavaScript without purchasing a license
- Client Side Script

(Reference: W3Schools Webpage, http://www.w3schools.com/js/js_intro.asp )
PHP

- PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML

- Server Side Script

- Supports variety of database types including MySQL
Database Structure

Database

Indiana Water Monitoring Info
(Latitude, Longitude, Name, Agency, Agency Type, Site Number, Address, parameter type, parameter, frequency, publicity, recording period, contact, Hydrologic Unit code, Login ID)

Login Information
(Login ID, email, Name, agency, Address)

Agency Information
(Sub agencies According to agency Type)
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agency</td>
<td>Agency name</td>
<td>Higher Agency</td>
<td>Super Agency</td>
</tr>
<tr>
<td>Name</td>
<td>Dataset name</td>
<td>Quality</td>
<td>Data quality info availability</td>
</tr>
<tr>
<td>Site_no</td>
<td>Site number</td>
<td>Contact</td>
<td>Contact URL</td>
</tr>
<tr>
<td>Address</td>
<td>The location of data</td>
<td>Latitude</td>
<td>Latitude of coordinate (WGS 1984)</td>
</tr>
<tr>
<td>ParType</td>
<td>Parameter type (Flow, GW level, and etc)</td>
<td>longitude</td>
<td>Longitude of coordinate (WGS 1984)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Parameters measured</td>
<td>User_info</td>
<td>User Account</td>
</tr>
<tr>
<td>Frequency</td>
<td>Monitoring frequency</td>
<td>HUC12</td>
<td>Hydrologic Unit Code</td>
</tr>
<tr>
<td>Public</td>
<td>Public availability</td>
<td>HUC10</td>
<td>Hydrologic Unit Code</td>
</tr>
<tr>
<td>Earliest</td>
<td>Date of record starting</td>
<td>HUC08</td>
<td>Hydrologic Unit Code</td>
</tr>
<tr>
<td>Latest</td>
<td>Date of record finishing</td>
<td>Id</td>
<td>Point id</td>
</tr>
</tbody>
</table>
Programming Overview

Logout.php

Index.php
(Main page)

Logged.php

Login.php
Embedding GoogleMaps

- `<script src='http://maps.google.com/maps?file=api&amp;v=2&amp;key=xxxxx' type='text/javascript'></script>`


```javascript
function load() {if (GBrowserIsCompatible()) {
  var map = new GMap2(document.getElementById('map'));
  map.addMapType(G_PHYSICAL_MAP);
  map.addControl(new GLargeMapControl());
  map.addControl(new GHierarchicalMapTypeControl());
  map.addControl(new GScaleControl());
}
```

- Inside HTML; `<div id='map' style=' height: 600px'></div>`
Connecting to database & sending query

```php
$md = mysql_connect(server, id, password);
mysql_select_db(database, $md);
$query = "SELECT Name, latitude, longitude, … FROM table WHERE Name LIKE '%".$_GET["variable_1"]."%' AND latitude=$_GET["variable2"] AND longitude=$_GET["variable_3"]…;
$qresult = mysql_query($query) or die();
while($row=mysql_fetch_row($qresult)){
    $name[$i]=$row[0]; $latitude[$i]=$row[1];
    $longitude[$i]=$row[2];
    $agency[$i]=$row[3];
    .....
```
Passing PHP Variable to Javascript

- var point = new GLatLng('.$lat[$j].', '.$lon[$j].');

- map.addOverlay(createMarker(point, '.$agency[$j].', '.$agency_type[$j].', '.$name[$j].', ...));

- function createMarker(point, Agency, agency_type, name, ...) {
  var icon = new GIcon(baseIcon);
  if (index==0) {icon.image = "red.png"; }
  else if (index==1) {icon.image = "blue.png"; }
  var marker = new GMarker(point, icon);
Displaying Information

Agency: The Nature Conservancy
Dataset: Darter Surveys
Location: West Branch of Mosquito Creek
Site #:
Parameter(s) sampled: community bioassessment
Parameter Type: Fish, Macroinvertebrates
Monitoring Frequency: triennial
Publicly Available?: No
Data Quality Information: The Nature Conservancy
Date of record: 2007 to On-going
HUC12: 051401040101
For more information:
http://www.nature.org/wherewework/northamerica/states/ir

Agency: The Nature Conservancy
Dataset: Blue River Project Water Quality Monitoring
geocoder = new GClientGeocoder();

function showAddress(address) {
    if (geocoder) {
        geocoder.getLatLng(address, function(point) {
            if (!point) { alert(address + ' not found');}
            else {map.setCenter(point, 13); }});
    }
}
if ($_GET["down"]=='Yes' or
$_GET["down_all"]=='Yes')
{$filename = 'download.xls';
header("Pragma: public");header("Expires: 0");
header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
header("Content-Type: application/force-download");
header("Content-Type: application/octet-stream");
header("Content-Type: application/download");
header("Content-Disposition: attachment; filename=".basename($file).";");
header("Content-Transfer-Encoding: binary");
header("Content-Length: ".filesize($file));
readfile($filename);}
Log in.

- Create Session variable for loginID
  ```php
  session_start(); // start the session
  header("Cache-control: private");
  $_SESSION["name"] = $login_id;
  header("location: logged.php");
  ```

- Data holders manage their data by themselves
Logged.php

- Loads data to database
- SQL statement

$_query = "INSERT INTO table (column1, column2, ... ) VALUES ("."$_GET["parameter1"].", ",".$_GET["parameter2"]', ...);"
### Editing Data

- **Edit**
  ```php
  $query = "UPDATE table SET column='".$_GET[variable]."'… WHERE ID='".$_GET['variable_ID']."' and user='".$_SESSION['variable_session']."""
  ```

- **Delete**
  ```php
  $query = "DELETE FROM table WHERE ID='".$_GET['variable_ID']."' and user='".$_SESSION['variable_session']."""
  ```
Conclusion

- Indiana water monitoring inventory was created by Googlemaps API, Javascripts, PHP and Mysql

- Water monitoring information from variety of sources are stored

- User can find the water monitoring information conveniently by GUI

- Data holders can upload, edit and delete information

- User can download the information as text based file