

Implementation of an interactive database interface utilizing HTML, PHP, JavaScript, and MySQL in support of water quality assessments in the Northeastern North Carolina Pasquotank Watershed

Antonio Guion, Kathryne Burton, Hagen Hodgkins, Derek Morris Jr., Mentor Mr. Jeff Wood
Center of Excellence in Remote Sensing Education and Research
Elizabeth City State University
Elizabeth City, North Carolina, USA

Abstract—The Center of Excellence in Remote Sensing Education and Research (CERSER) has implemented three research projects during the summer Research Experience for Undergraduates (REU) program gathering water quality data for local waterways. The data has been compiled manually utilizing pen and paper and then entered into a spreadsheet. With the spread of electronic devices capable of interacting with databases, the development of an electronic method of entering and manipulating the water quality data was pursued during this project.

This project focused on the development of an interactive database to gather, display, and analyze data collected from local waterways. The database and entry form were built in MySQL on a PHP server allowing participants to enter data from anywhere Internet access is available. This project then researched applying this data to the Google Maps site to provide labeling and information to users. The NIA server at <http://nia.ecsu.edu> is used to host the application for download and for storage of the databases.

This project was built on water quality data gathered by the Summer Research Experience for Undergraduate teams during the summers of 2014, 2013, and 2011.

Index Terms— HTML, JavaScript, PHP, Database, MySQL, Water quality, Pasquotank, watershed, data visualization

I. INTRODUCTION

The Summer Undergraduate Research Experience program sponsored by the Center of Excellence in Remote Sensing Education and Research (CERSER) at Elizabeth City State University has run three summers of water collection and analyzation in the Pasquotank Watershed. The 2011, 2013, and 2014 teams collected over 350 samples from the Pasquotank River and four tributaries. These samples included the following water tests: dissolved oxygen, turbidity, clarity, total dissolved solids (TDS), salinity, pH, and conductivity [1][2][3]. Several pieces of this data have a direct correlation to sea level rise where coastal estuaries experiencing a rise in their levels show an increase in salt level indicators [3]. Other data such as date, position, and time were also included in spreadsheets created by Microsoft

Excel. The data was then coded into HTML to be displayed in Google Maps. Here the position was visually represented and the data was displayed in an information balloon.

The goals of this project were to combine and convert the spreadsheet data into a MySQL database, create data display, entry, and modification pages in PHP and HTML, and to finally provide a dynamic approach to the Google Maps display allowing users to select the data to be displayed. These goals will form the basis for continued refinement toward mobile development allowing data to be input in the field during collection.

II. SOFTWARE/LANGUAGES

A. Overview

There were several pieces of software utilized for the completion of this project. These ranged from a database management system (DBMS) to a graphical user interface (GUI) for developing basic and advanced web pages. Several languages were also brought together to store, manipulate, and visualize the data. A brief overview of the software and languages used follows.

B. HTML

Hypertext Markup Language (HTML) is the framework for labeling content documents to achieve font, color, graphic and hyperlink effects on Internet pages. HTML is used to make website pages. The Web browsers can read HTML records and create them into visible or audible web pages. The browsers do not show the HTML labels and scripts, but utilize them to interpret the substance of the page [5].

HTML comprises of a series of short codes written into a content document by the sited author; these are the tags. The text is then spared as a HTML file and viewed through a program, similar to Internet Explorer or Netscape Navigator. This program reads the files and makes the translation text into a visible structure, ideally rendering the page as the author had planned. Writing your own HTML involves utilizing tags. You can utilize anything from a simple text-editor tool to an intense graphical manager to make a HTML page [4].

C. Google Maps

Google Map is a mobile web mapping application that is provided by Google. Google Map offers the use of satellite visualization, road maps, and functions such as a route planner for traveling. Google Maps implements the use of street view, which incorporates the use of photographs taken from vehicles. Google Maps for Mobile grants the user the ability to use Global Positioning System (GPS) using data from wireless and cellular networks. Google Maps is capable of being used on third-party websites [9][10].

The team is using Google Maps to track and reference the location of the trips taken by the watershed teams from 2012-2014. The data taken from the trips are converted into databases.

D. phpMyAdmin

phpMyAdmin is an open source database tool that joins Structured Query Language (SQL) with the internet. The tool uses PHP script to perform a variety of database functions. phpMyAdmin supports a wide range of operations on MySQL, MariaDB and Drizzle. The database tool can also perform the user interface, while retaining the ability to directly execute any SQL statements [11].

Throughout this project phpMyAdmin was utilized to construct a database that would further be used to create an XML file, which Google Maps would be able to read.

E. Dreamweaver

Adobe Dreamweaver is a proprietary web development tool developed by Adobe Systems. Created by Macromedia in 1997, Adobe Dreamweaver is used to create and design webpages through JavaScript and HTML. This application also allows the user to edit text through the "What You See Is What You Get" (WYSIWYG) method [12].

Throughout this project, Dreamweaver was used to document the project and to create PHP and HTML files that can access manipulate and analyze the database. This created a visual representation in Google Maps of the database.

F. PHP

PHP, or PHP Hypertext Preprocessor, is an open source scripting language that is tailored for use in website development and can be included within HTML. It is similar to asking the server a math question and receiving the answer in return without the server revealing how the answer was reached.

PHP was created in 1994 by Rasmus Lerdorf originally it was a non-released version that was used to track visitors to his online resume. In 1995 a version was released to others under the title of Personal Home Page Tools. This version consisted of a simple parser and several basic macros.

PHP is supported by every major Operating system and supports many database programs. (Adabas D, InterBase, Solid, dBase, mSQL, Sybase, Empress, MySQL, Velocis, FilePro, Oracle, Unix dbm, Informix, PostgreSQL) [6][7].

G. MySQL

MySQL is an open source relational database management system based on the structure query language (SQL). It is used for creating and deleting databases and their associated parts while also manipulating information in the database. Many of the SQL typical commands, such as ADD, DROP, INSERT, and UPDATE are used with MySQL.

While mainly found on web servers, MySQL can be used for a range of applications. Its use with a web page allows it to be "dynamic." In other words, the information on the page changes depending on what is retrieved from the database. These web pages make up a database-driven web site [8].

This project used the MySQL commands to retrieve filtered data from the database and then order it by the ID field.

H. XML

Extensible Markup Language (XML) is a markup dialect that characterizes a set of principles for encoding records in an organization which is both comprehensible and machine-readable. The XML standard is an adaptable technique to make data configurations and electronically impart organized information thru general society web and thru corporate system. XML can back settled components, and components inside components. This permits XML to help progressive structures. Component names portray the substance of the component, and the structure depicts the relationship between the components [13].

I. JavaScript

JavaScript is a cross-platform scripting program that provides web pages with interactivity on most browsers. It was originally developed under the name LiveScript by Netscape. JavaScript is an object-oriented language with detailed scripting codes.

As with other programming languages, JavaScript has certain key concepts and phrases that control the interaction with the web page. Four primary functions of JavaScript that need to be studied for successful use include objects, methods, events, and functions [14][15].

This project utilized JavaScript to manipulate the data in order that it could be visualized. Functions and events were used to interact with Google Maps and provide the data in the correct format.

III. METHODOLOGY

A. Combine Spreadsheets

Research teams from 2011, 2013, and 2014 entered their data into Excel spreadsheets in order to create HTML files for analyzation and data visualization. These sheets varied slightly as new tests (turbidity) were added in 2013 and 2014. Units of measurement also had to be standardized before the files were combined (micro, milli). Wind direction measurements on calm days were entered as "0" or "calm" for several entries and were changed to a null entry to reflect the lack of direction.

Another issue that arose was the display of unit symbols for the position of the sample points. Hours (°), minutes (′), and seconds (″) were often not written correctly and did not translate to the database and XML files correctly. This

they were to hold (text, integers, and floating decimals). The screenshot of the database structure can be seen in Fig. 1.

The next step was to import the data from the combined

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(5)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	SRCNAME	varchar(60)	latin1_swedish_ci		Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	DATE	varchar(12)	latin1_swedish_ci		Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	PT	varchar(5)	latin1_swedish_ci		Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	TIME	varchar(5)	latin1_swedish_ci		Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	LATHRS	varchar(3)	latin1_swedish_ci		Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
7	LATMIN	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
8	LATSEC	float(4,2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
9	LONGHRS	varchar(3)	latin1_swedish_ci		Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
10	LONGMIN	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
11	LONGSEC	float(4,2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
12	GLAT	float(10,6)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
13	GLONG	float(10,6)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
14	DOX	float(5,2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
15	CDOX	float(6,2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
16	PH	float(3,1)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
17	WTEMP	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
18	ATEMP	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
19	CLAR1	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
20	CLAR2	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
21	CAVG	float(4,1)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
22	TUR1	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
23	TUR2	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
24	TAVG	float(4,1)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
25	TDS	int(4)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
26	WKTS	int(2)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
27	WDIR	int(3)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
28	SALT	int(4)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
29	COND	float(6,3)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Figure 1 Screenshot of the database structure

problem was alleviated by splitting the positions into three separate fields, HOURS, MINUTES, and SECONDS, and then reintroducing the symbols when the data was placed in the working files. This will make data entry and retrieval easier as researchers will only be concerned with the actual numbers and not the symbols.

Once these issues were resolved, the five files (2013 and 2014 included two trips to each waterway), were compiled into one. This process was repeated many times in order to provide an accurate resource to import into the database.

B. Database

Using phpMyAdmin, a graphical interface for MySQL, the database structure was created. A single table with 28 data fields and one key field was created. The fields were given their individual characteristics depending on the data

spreadsheets. The data was first exported from the Excel file into a text file in the Comma Separated Value (CSV) format. The CSV file was then opened in a text editor where the last paragraph mark was removed. The “Save As” function was used again in order to keep the CSV format (.csv) and not convert it to a straight text file format (.txt). This process was repeated many times in order to remove errors in the data that were found after the import.

C. Data Pages

In order to view the data without the use of phpMyAdmin, a data display page was created using Dreamweaver (Fig. 2). This page displays the data within the database in table form ordered by the ID number of each record. The display page uses the connection file parameters to connect with the database.

	ID	SRCNAME	DATE	PT	TIME	LAT	LONG	GLAT	DOX	CDOX	PH	WTEMP	ATEMP	CLAR1	CLAR2
Modify Delete	1	Newbegun Creek	6/4/14	N1	11:20	36°12'47.31"	76°10'22.32"	36.213139	76.172859	1.10	7.4	24	35	21	18
Modify Delete	2	Newbegun Creek	6/4/14	N2	11:08	36°12'39.47"	76°10'26.87"	36.210945	76.174141	1.20	7.5	24	35	23	19
Modify Delete	3	Newbegun Creek	6/4/14	N3	10:55	36°12'31.85"	76°10'23.76"	36.208866	76.173279	1.60	7.6	24	35	27	18
Modify Delete	4	Newbegun Creek	6/4/14	N4	10:35	36°12'24.23"	76°10'6.39"	36.206730	76.168442	2.80	7.8	24	35	24	25
Modify Delete	5	Newbegun Creek	6/4/14	N5	10:28	36°12'36.37"	76°9'52.63"	36.210102	76.164619	4.00	7.7	25	35	25	24
Modify Delete	6	Newbegun Creek	6/4/14	N6	10:21	36°12'33.41"	76°0'32.34"	36.209278	76.158974	5.30	8.0	25	31	23	23
Modify Delete	7	Newbegun Creek	6/4/14	N7	10:14	36°12'44.30"	76°9'32.40"	36.212307	76.158997	4.80	8.0	25	30	26	21
Modify Delete	8	Newbegun Creek	6/4/14	N8	9:46	36°13'14.50"	76°7'51.40"	36.220695	76.130943	7.20	8.3	26	29	20	20
Modify Delete	9	Newbegun Creek	6/4/14	N9	10:00	36°13'12.61"	76°8'51.61"	36.220169	76.147667	3.90	8.2	25	31	21	22
Modify Delete	10	Newbegun Creek	6/4/14	N9a	9:52	36°13'18.40"	76°8'18.60"	36.221779	76.138496	4.20	8.3	25	28	30	21
Modify Delete	11	Newbegun Creek	6/4/14	N10	10:05	36°12'52.00"	76°8'58.10"	36.214443	76.149475	7.40	8.1	26	30	23	22
Modify Delete	12	Newbegun Creek	6/4/14	N11	9:37	36°13'0.00"	76°7'21.00"	36.216667	76.122498	6.40	8.4	25	29	26	18
Modify Delete	13	Mill Dam Creek	6/2/14	M1	8:50	36°18'35.10"	76°7'51.10"	36.309750	76.130859	0.20	6.8	21	21	14	15
Modify Delete	14	Mill Dam Creek	6/2/14	M2	9:20	36°18'18.90"	76°7'48.60"	36.305248	76.130165	0.40	6.8	21	22	25	25
Modify Delete	15	Mill Dam Creek	6/2/14	M3	9:41	36°18'3.40"	76°7'54.90"	36.300945	76.131897	0.70	6.7	23	24	22	22
Modify Delete	16	Mill Dam Creek	6/2/14	M4	9:58	36°17'47.00"	76°8'2.40"	36.296391	76.134003	1.60	6.9	23	27	17	19
Modify Delete	17	Areneuse Creek	6/4/14	A1	16:58	36°18'1.80"	76°6'55.70"	36.300499	76.115471	4.30	7.5	26	25	29	29
Modify Delete	18	Areneuse Creek	6/4/14	A2	12:45	36°17'53.60"	76°7'4.70"	36.298222	76.117973	2.00	7.2	27	43	20	22

Figure 2 Data show file created in Dreamweaver

Each record has two links, “Modify” and “Delete,” which enable the user to either modify the current record or delete it. The modify link takes the user to a form page that is prefilled with the current record (Fig. 3). The user can then save the modifications and is returned to the data display page. The delete link deletes the current record.

A Data Entry page was created so that new records could be entered. The point and latitude/longitude positions can be entered ahead of the research trip to the waterway. The other data collected in the field can be entered through the modification of each record from the data view page.

ID: 1

SRCNAME:

DATE:

DD/MM/YY

PT:

TIME:

LAT:

LONG:

GLAT:

GLONG:

DOX:

CDOX:

PH:

WTEMP:

ATEMP:

CLAR1:

CLAR2:

CAVG:

TUR1:

TUR2:

TAVG:

TDS:

WKTS:

WDIR:

SALT:

COND:

Figure 3 Screenshot of data modification page

IV. VISUALIZATION OF DATA

A. Overview

Once the database was established, the next step in the process was to create a visualization of the data in Google Maps. The basic steps are to pass the requested parameters to the database, convert the retrieved data into an XML format, apply the data to JavaScript variables, and then utilize Google Map functions to create points with the retrieved data onto a map. A screenshot of the current interface can be found in Appendix 1. The file can be viewed online at: http://nia.ecsu.edu/googleDOM_example.html.

B. XML builder file

The requested data must be converted to an XML format in order that Google Maps can display it. This is accomplished by calling a separate file, in this project, the DOM_example.php file, which utilizes PHP and MySQL commands to import and format the data into XML format. The first step in building the XML file for Google Maps is to retrieve the PHP parameters passed in the URL link from the GoogleDom_example.html file. For this project the river/creek names and the testing dates are being passed to receive a limited number of points. This data is contained within the variables SRCNAME and DATE. The data are retrieved using the PHP command “\$_GET” and then applying them to variables. The variables are used to form a MySQL statement to retrieve the data from the database. The statement takes the following format:

```
"SELECT * FROM pqt_table2 WHERE DATE=%s
AND SRCNAME=%s ORDER BY 'ID' ASC"
```

This statement retrieves all of the records from the table “pqt_table2” if the date and source fields match the previously retrieved variables. The records are then ordered by the field “ID” in ascending order.

Once the data has been imported, it is processed through a loop function which echoes out the data in XML format. An example of this XML file is found in Fig. 4 utilizing the following URL with PHP variables:

```

▼<markers>
<marker SRCNAME="Sawyers Creek" DATE="6/27/13" PT="S1" TIME="9:48" LATHRS="36" LATMIN="21" LATSEC="1.73" LONGHRS="76"
LONGMIN="9" LONGSEC="49.88" GLAT="36.350479" GLONG="-76.163857" DOX="0.70" CDOX="0.70" PH="6.9" WTEMP="26" ATEMP="28"
CLAR1="14" CLAR2="14" CAVG="14.5" TUR1="5" TUR2="4" TAVG="4.5" TDS="640" WKTS="0" WDIR="0" SALT="460" COND="0.924"/>
<marker SRCNAME="Sawyers Creek" DATE="6/27/13" PT="S2" TIME="10:05" LATHRS="36" LATMIN="20" LATSEC="46.40" LONGHRS="76"
LONGMIN="9" LONGSEC="45.70" GLAT="36.346222" GLONG="-76.162697" DOX="0.60" CDOX="0.60" PH="6.8" WTEMP="27" ATEMP="29"
CLAR1="20" CLAR2="20" CAVG="20.5" TUR1="5" TUR2="4" TAVG="4.5" TDS="710" WKTS="0" WDIR="0" SALT="510" COND="1.027"/>
<marker SRCNAME="Sawyers Creek" DATE="6/27/13" PT="S3" TIME="10:20" LATHRS="36" LATMIN="20" LATSEC="33.60" LONGHRS="76"
LONGMIN="9" LONGSEC="54.30" GLAT="36.342667" GLONG="-76.165085" DOX="1.10" CDOX="1.10" PH="6.5" WTEMP="26" ATEMP="30"
CLAR1="18" CLAR2="18" CAVG="18.5" TUR1="6" TUR2="5" TAVG="5.5" TDS="740" WKTS="0" WDIR="0" SALT="530" COND="1.061"/>
<marker SRCNAME="Sawyers Creek" DATE="6/27/13" PT="S4" TIME="10:31" LATHRS="36" LATMIN="20" LATSEC="23.90" LONGHRS="76"
LONGMIN="10" LONGSEC="7.30" GLAT="36.339973" GLONG="-76.168694" DOX="1.60" CDOX="1.60" PH="6.6" WTEMP="27" ATEMP="28"
CLAR1="16" CLAR2="16" CAVG="16.5" TUR1="5" TUR2="4" TAVG="4.5" TDS="750" WKTS="3" WDIR="210" SALT="530" COND="1.075"/>

```

Figure 4 DOM_example.php XML output

“http://nia.ecsu.edu/dom_example.php?DATE=6/27/13&SRCNAME=Sawyers%20Creek”

This data is then passed back to the calling file, in this case, GoogleDOM_example.html, to create the map.

C. Google Maps Display file

The file GoogleDOM_example.html is used to display the selected records from the database in Google Maps. The file first creates the “load” function which will display the map with parameters such as zoom level, map type (“road map” in this case), and the map center coordinates as seen in Fig. 5.

```

//This is the main function for creating the map from the XML data
function load() {
    var map = new google.maps.Map(document.getElementById("map"), {
        center: new google.maps.LatLng(36.2965, -76.2161),
        zoom: 12,
        mapTypeId: 'roadmap'
    });
}

```

Figure 5 Load function in JavaScript

The next step is to retrieve the PHP variables attached to the URL as previously shown using the PHP “\$_GET” function. These variables are then attached to the relative URL “DOM_example.php” which is then called to retrieve the data and format it in XML format as previously outlined.

The data is then assigned to JavaScript variables while looping through each record retrieved. The latitude and longitude data contained in the fields GLAT and GLONG are combined into one variable (point) to be used as a definition for the Google Maps marker location. This format is the signed degrees format (DDD.dddd). For example, the standard latitude/longitude format is 36° 21' 1.73"N, 76° 09' 49.88"W and the signed degrees format would be 36.350479, -76.163857 for the same position.

These variables are then combined with HTML code and placed into the variable HTML. This variable is used to populate the “information balloon” when a user clicks on a marker. An example of this is shown in Fig. 6.

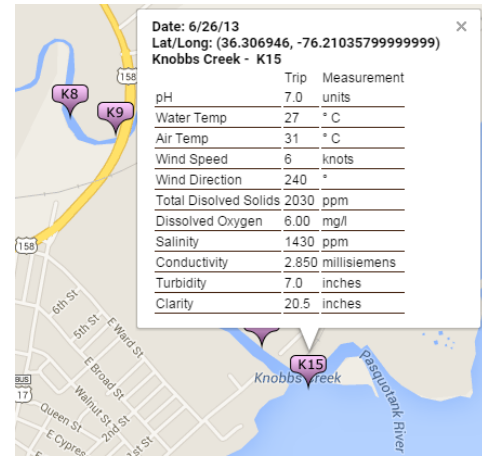


Figure 6 Information balloon with data

At the end of the variable assignment, the location of the marker graphic is placed into the variable “icon.” This location is a URL to a portable network graphic (PNG) file utilizing the variable PT.

A “listening event” is created so that the user can click on the marker to view the data for that location. This data is placed in the information balloon described above.

The GoogleDOM_example.html file has been formatted with links passing the needed variables for each research trip taken during the three years of research.

V. RESULTS

The construction and implementation of the various files and databases were completed with minimal roadblocks. The compilation of the spreadsheets into a singular database proved to be a tedious task due to minor inconsistencies in the spreadsheets. Ensuring that the CSV files were in the correct format enabled the project to continue smoothly.

The data show, modify, entry, and delete files were quickly developed due to the features within Dreamweaver. These visual features included database connection and form building, which made the process smoother. The DOM_example.php and the GoogleDOM_example.html files were more intricate due to the amount of hand coding that had to be done. Knowledge of PHP, JavaScript, and MySQL were greatly increased due to the amount of time spent in these languages. A programming tool used during this development was the echoing/printing of variable

contents, which allowed the contents of variables to be followed during the process.

VI. CONCLUSION

The creation of these files, along with the compilation of the spreadsheets, will enhance the ability of research students gathering data in the field to do so more accurately. The continued development of the visualization tool allows a greater ability to analyze the results and compare them with previous data.

Now that the framework has been laid for the data flow, further enhancements can be added to increase the amount of available data and how it is displayed. This will allow researchers to further track the effects of climate change on local watersheds.

VII. FUTURE WORKS

There are several items that will aid in the continued development of this database and the increased value it will have in gathering data. The first recommended future work is to reformat the date data from MM/DD/YY to YYYY-MM-DD in order to assist the sorting of records by the date the sample was taken.

The next step is to create an HTML form enabling the user to perform specific searches on the database. This page would allow the user to perform a variety of queries depending on what was being analyzed.

The data show, input, and modification files should be further developed for use as mobile applications using either HTML5 for a web based application or a native mobile language. This would further assist the researchers to enter data from field research into the database.

Now that the basic pages have been coded, a cohesive design should be sought to make the more user friendly and attractive. Standard Cascading Style Sheet (CSS) formats should be implemented to achieve this affect.

The DELETE link on the Data Show page needs a pause moment where the user can have a second chance to decide if they want to delete a record. Accidents do happen and the user needs the chance to correct a mistaken click.

VIII. ACKNOWLEDGEMENTS

We would like to acknowledge the assistance of Dr. Linda Hayden who has provided the funding and the opportunity for this projects completion.

REFERENCES

- [1] McKenzie, R., Stevenson, J., Wood, J., Hayden, L. (2014), "A comparative study to the 2011/2013 water quality assessments in the Pasquotank Watershed in Northeastern North Carolina with a sea level rise component." [Online]. Available. <http://nia.ecsu.edu/reuomps2014/teams/water/>.
- [2] Brice, D., Brice, R., Lynn, A., Wood, J., Hayden, L. (2013), "A comparative study of the 2011/2013 water quality assessments in the Pasquotank Watershed in Northeastern North Carolina." [Online]. Available. <http://nia.ecsu.edu/reuomps2013/teams/water/>.
- [3] Deese, A., Hassell, S., Jones, A., Wood, J., Hayden, L. (2011), "Water-Quality Assessment of the Pasquotank River Watershed; Analysis of Dissolved Oxygen, pH, Salt, Total Dissolved Solids, and Conductivity." [Online]. Available. <http://nia.ecsu.edu/ureomps2011/teams/ptank/>.
- [4] Refsnes Data-W3Schools, (1999-2015), "Introduction to HTML." [Online]. Available. http://www.w3schools.com/html/html_intro.asp.
- [5] Shannon, R., (2000-2015), "What is HTML." [Online]. Available. <http://www.yourhtmlsource.com/starthere/whatishtml.html>.
- [6] The PHP Group, (2001-2015), "PHP: What is PHP." [Online]. Available. <http://www.yourhtmlsource.com/starthere/whatishtml.html>.
- [7] Bakken, S., et al., Virginia Tech Digital Library and Archives, (2001-2015), "PHP Manual." [Online]. Available. <http://scholar.lib.vt.edu/manuals/php3.0.6/manual.html>.
- [8] TechTerms.com (2007). "MySQL." [Online]. Available. <http://techterms.com/definition/mysql>.
- [9] Rouse, M. (2013). "What is Google Maps?." [Online]. Available. <http://whatis.techtarget.com/definition/Google-Maps>.
- [10] Google Developers. (2015). "Google Maps Developer Documentation." [Online]. Available. <https://developers.google.com/maps/documentation/>.
- [11] phpMyAdmin. (2015). "Bringing MySQL to the web." [Online]. Available. <http://www.phpmyadmin.net/>.
- [12] Lynda.com. (1995-2015). "What is Dreamweaver?." [Online]. Available. <http://www.lynda.com/Dreamweaver-tutorials/What-Dreamweaver/124084/138659-4.html>.
- [13] Doszkocs, T., et al. (2001-2015). "XML (Extensible Markup Language)." [Online]. Available. <http://searchsoa.techtarget.com/definition/XML>.
- [14] Crockford, D. (2001). "JavaScript: The World's Most Misunderstood Programming Language." [Online]. Available. <http://javascript.crockford.com/javascript.html>.
- [15] Free, C. (2004). "JavaScript: An Evaluation of a Web-Related Technology - Description." [Online]. Available. <http://www.usca.edu/free/javascript/descrip.htm>.

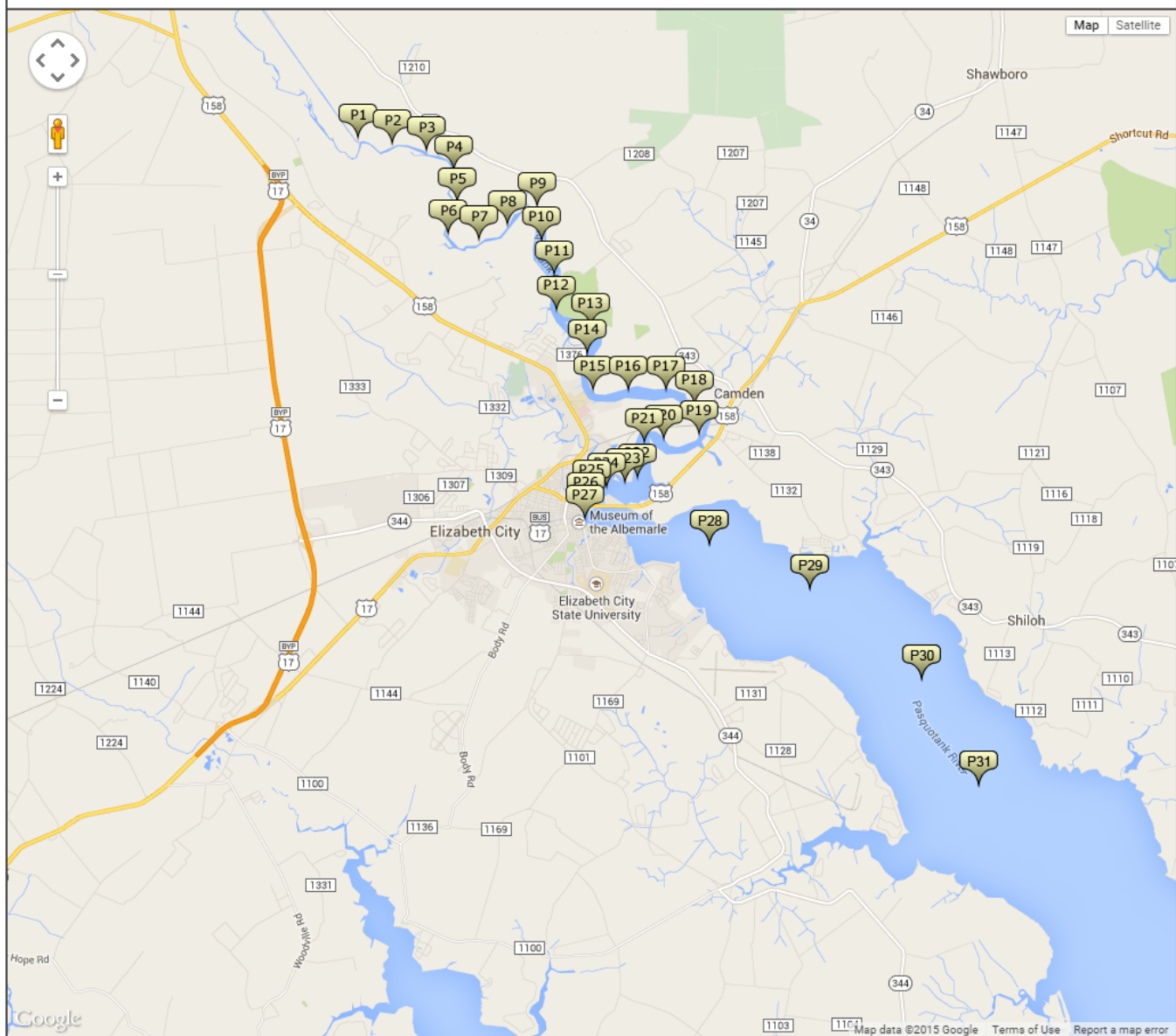
APPENDIX 1 GOOGLEDOM_EXAMPLE.HTML SCREENSHOT

2015 Multimedia Team

Implementation of an interactive database interface utilizing HTML, JavaScript, PHP, and MySQL in support of water quality assessments in the Northeastern North Carolina Pasquotank Watershed

[BACK TO SITE](#)

Knobbs Creek - 6/17/11	Sawyers Creek - 6/16/11	Pasquotank River - 6/15/11	Areneuse Creek - 6/14/11	Newbegun Creek - 6/30/11
			Mill Dam Creek - 6/14/11	Newbegun Creek - 6/7/11
				Newbegun Creek - 6/13/11
Knobbs Creek - 6/20/13	Sawyers Creek - 6/13/13	Pasquotank River - 6/18/13	Areneuse Creek - 6/21/13	Newbegun Creek - 6/19/13
Knobbs Creek - 6/26/13	Sawyers Creek - 6/27/13	Pasquotank River - 6/24/13	Areneuse Creek - 6/14/13	Newbegun Creek - 6/24/13
			Mill Dam Creek - 6/21/13	
			Mill Dam Creek - 6/14/13	
Knobbs Creek - 6/9/14	Sawyers Creek - 6/6/14	Pasquotank River - 6/4/14	Areneuse Creek - 6/4/14	Newbegun Creek - 6/4/14
Knobbs Creek - 6/17/14	Sawyers Creek - 6/11/14	Pasquotank River - 6/20/14	Areneuse Creek - 6/10/14	Newbegun Creek - 6/20/14
			Mill Dam Creek - 6/2/14	
			Mill Dam Creek - 6/10/14	



APPENDIX 2 DOM_EXAMPLE.PHP

```

1  <?phprequire_once('Connections/PQWS_DB.php'); ?>
2
3  <?php
4  if (!function_exists ("GetSQLValueString" )) {
5  function GetSQLValueString ($theValue, $theType, $theDefinedValue = "", $theNotDefinedValue = "")
6  {
7      if (PHP_VERSION < 6) {
8          $theValue = get_magic_quotes_gpc () ? stripslashes ($theValue) : $theValue;
9      }
10
11     $theValue = function_exists ("mysql_real_escape_string" ) ? mysql_real_escape_string ($theValue) : m
        ysql_escape_string ($theValue);
12
13     switch ($theType) {
14         case "text" :
15             $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL" ;
16             break;
17         case "long" :
18         case "int" :
19             $theValue = ($theValue != "") ? intval ($theValue) : "NULL" ;
20             break;
21         case "double" :
22             $theValue = ($theValue != "") ? doubleval ($theValue) : "NULL" ;
23             break;
24         case "date" :
25             $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL" ;
26             break;
27         case "defined" :
28             $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue ;
29             break;
30     }
31     return $theValue;
32 }
33 }
34
35 //Retrieve PHP Variables from URL
36 $colname_Recordset1 = "-1";
37 if (isset ($_GET['DATE'])) {
38     $colname_Recordset1 = $_REQUEST['DATE'];
39 }
40
41 $colname2_Recordset1 = "-1";
42 if (isset ($_GET['SRCNAME'])) {
43     $colname2_Recordset1 = $_GET['SRCNAME'];
44 }
45
46 mysql_select_db ($database_PQWS_DB, $PQWS_DB);
47 $query_Recordset1 = sprintf ("SELECT * FROM pqt_table2 WHERE DATE=%s AND SRCNAME=%s ORD
        ER BY 'ID' ASC", GetSQLValueString ($colname_Recordset1, "text"), GetSQLValueString ($colname2_R
        ecordset1, "text" ));
48 $Recordset1 = mysql_query ($query_Recordset1, $PQWS_DB) or die(mysql_error ());
49 $totalRows_Recordset1 = mysql_num_rows ($Recordset1 );
50
51 /*
52 //MANUAL MANIPULATION OF DATA QUERY
53 mysql_select_db($database_PQWS_DB, $PQWS_DB);

```



```

54 $query_Recordset1 = "SELECT * FROM pqt_table2 WHERE DATE='2011-06-15' AND SRCNAME='Pasqu
otank River' ORDER BY ID ASC";
55 $Recordset1 = mysql_query($query_Recordset1, $PQWS_DB) or die(mysql_error());
56 $row_Recordset1 = mysql_fetch_assoc($Recordset1);
57 $totalRows_Recordset1 = mysql_num_rows($Recordset1);
58 */
59
60 function parseToXML ($htmlStr )
61 {
62 $xmlStr =str_replace ('<','&lt;',$htmlStr );
63 $xmlStr =str_replace ('>','&gt;',$xmlStr );
64 $xmlStr =str_replace ('"' , '&quot;' , $xmlStr );
65 $xmlStr =str_replace ('"' , '&#39;' , $xmlStr );
66 $xmlStr =str_replace ('&' , '&amp;' , $xmlStr );
67 return $xmlStr ;
68 }
69
70 header ("Content-type: text/xml; charset=iso-8859-1" );
71
72 // Start XML file, echo parent node
73 echo '<markers>' ;
74
75 //TroubleShooting
76 //$countme = 0;
77
78 // Iterate through the rows, printing XML nodes for each
79 while ($row = mysql_fetch_assoc ($Recordset1 )){
80 // ADD TO XML DOCUMENT NODE
81 // echo $countme;
82 echo '<marker ' ;
83 echo 'SRCNAME="' . parseToXML ($row['SRCNAME']) . '" ' ;
84 echo 'DATE="' . parseToXML ($row['DATE']) . '" ' ;
85 echo 'PT="' . parseToXML ($row['PT']) . '" ' ;
86 echo 'TIME="' . parseToXML ($row['TIME']) . '" ' ;
87 echo 'LATHRS="' . $row['LATHRS'] . '" ' ;
88 echo 'LATMIN="' . $row['LATMIN'] . '" ' ;
89 echo 'LATSEC="' . $row['LATSEC'] . '" ' ;
90 echo 'LONGHRS="' . $row['LONGHRS'] . '" ' ;
91 echo 'LONGMIN="' . $row['LONGMIN'] . '" ' ;
92 echo 'LONGSEC="' . $row['LONGSEC'] . '" ' ;
93 echo 'GLAT="' . $row['GLAT'] . '" ' ;
94 echo 'GLONG="' . $row['GLONG'] . '" ' ;
95 echo 'DOX="' . $row['DOX'] . '" ' ;
96 echo 'CDOX="' . $row['CDOX'] . '" ' ;
97 echo 'PH="' . $row['PH'] . '" ' ;
98 echo 'WTEMP="' . $row['WTEMP'] . '" ' ;
99 echo 'ATEMP="' . $row['ATEMP'] . '" ' ;
100 echo 'CLAR1="' . $row['CLAR2'] . '" ' ;
101 echo 'CLAR2="' . $row['CLAR2'] . '" ' ;
102 echo 'CAVG="' . $row['CAVG'] . '" ' ;
103 echo 'TUR1="' . $row['TUR1'] . '" ' ;
104 echo 'TUR2="' . $row['TUR2'] . '" ' ;
105 echo 'TAVG="' . $row['TAVG'] . '" ' ;
106 echo 'TDS="' . $row['TDS'] . '" ' ;
107 echo 'WKTS="' . $row['WKTS'] . '" ' ;
108 echo 'WDIR="' . $row['WDIR'] . '" ' ;

```

```
109 echo 'SALT=' . $row['SALT'] . "'";
110 echo 'COND=' . $row['COND'] . "'";
111 echo '/>';
112 $countme = $countme + 1;
113 }
114
115 // End XML file
116 echo '</markers>';
117
118 mysql_free_result ($Recordset1);
119
120 ?>
121
```

```

1 <!DOCTYPE html >
2 <head>
3   <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
4   <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
5   <title>2015 Multimedia Team - Water Quality Data Display</title>
6   <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js"></script>
7
8
9   <!--This "style" places underlines in the table within the balloon-->
10  <style type="text/css">
11    .dataunderline {
12      border-bottom-width: thin;
13      border-bottom-style: solid;
14      border-bottom-color: #452513;
15    }
16  </style>
17
18  <script type="text/javascript">
19    <![CDATA[
20
21    //This is the main function for creating the map from the XML data
22    function load() {
23      var map = new google.maps.Map(document.getElementById("map"), {
24        center : new google.maps.LatLng(36.2965, -76.2161),
25        zoom: 12,
26        mapTypeId: 'roadmap'
27      });
28
29      var infoWindow = new google.maps.InfoWindow;
30
31      //var SEARCHPARAM =
32      var domqueryJS = "<?php echo'?DATE=?><?php echo $_GET['DATE']?><?php echo'&SRCNAME=?><?php echo $_GET['SRCNAME']?>";
33
34      // Import XML data from database by opening the file DOM.php and place values into JavaScript variables
35      downloadUrl("DOM_example.php"+domqueryJS, function(data) {
36        var xml = data.responseXML;
37        var markers = xml.documentElement.getElementsByTagName("marker");
38        for (var i = '0'; i < markers.length; i++) {
39          var SRCNAME = markers[i].getAttribute("SRCNAME");
40          var DATE = markers[i].getAttribute("DATE");
41          var PT = markers[i].getAttribute("PT");
42          var TIME = markers[i].getAttribute("TIME");
43          var LATHRS = markers[i].getAttribute("LATHRS");
44          var LATMIN = markers[i].getAttribute("LATMIN");
45          var LATSEC = markers[i].getAttribute("LATSEC");
46          var LONGHRS = markers[i].getAttribute("LONGHRS");
47          var LONGMIN = markers[i].getAttribute("LONGMIN");
48          var LONGSEC = markers[i].getAttribute("LONGSEC");
49          var POINT = new google.maps.LatLng(
50            parseFloat(markers[i].getAttribute("GLAT")),
51            parseFloat(markers[i].getAttribute("GLONG")));
52          var DOX = markers[i].getAttribute("DOX");
53          var CDOX = markers[i].getAttribute("CDOX");
54          var PH = markers[i].getAttribute("PH");
55          var WTEMP = markers[i].getAttribute("WTEMP");

```

```

56     var ATEMP = markers[i].getAttribute("ATEMP");
57     var CLAR = markers[i].getAttribute("CAVG");
58     var TUR = markers[i].getAttribute("TAVG");
59     var TDS = markers[i].getAttribute("TDS");
60     var WKTS = markers[i].getAttribute("WKTS");
61     var WDIR = markers[i].getAttribute("WDIR");
62     var SALT = markers[i].getAttribute("SALT");
63     var COND = markers[i].getAttribute("COND");
64     var CCC = "DDD";
65
66     //Place the information needed for the "information balloon" into the variable "html"
67     var html =
68     "<span class='txtinfoWindow'><b>Date: "
69     + DATE +
70     "<br/>Lat/Long: "
71     + POINT +
72     "<br/>"
73     + SRCNAME +
74     "&nbsp;-&nbsp;" +
75     + PT +
76     "</b><br/><table border='0'><tr><td>&nbsp;</td><td> Trip</td><td>Measurement</td></tr><tr><td class
s='dataunderline'>pH </td><td class='dataunderline'>"
77     + PH +
78     "</td><td class='dataunderline'>units</td></tr><tr><td class='dataunderline'>Water Temp </td><td clas
s='dataunderline'>"
79     + WTEMP +
80     "</td><td class='dataunderline'>&deg; C</td></tr><tr><td class='dataunderline'>Air Temp </td><td clas
s='dataunderline'>"
81     + ATEMP +
82     "</td><td class='dataunderline'>&deg; C</td></tr><tr><td class='dataunderline'>Wind Speed </td><td cl
ass='dataunderline'>"
83     + WKTS +
84     "</td><td class='dataunderline'>knots</td></tr><tr><td class='dataunderline'>Wind Direction </td><td cl
ass='dataunderline'>"
85     + WDIR +
86     "</td><td class='dataunderline'>&deg;</td></tr><tr><td class='dataunderline'>Total Dissolved Solids </td
><td class='dataunderline'>"
87     + TDS +
88     " </td><td class='dataunderline'>ppm</td></tr><tr><td class='dataunderline'>Dissolved Oxygen </td
><td class='dataunderline'>"
89     + DOX +
90     "</td><td class='dataunderline'> mg/l</td></tr><tr><td class='dataunderline'>Salinity </td><td class='da
taunderline'>"
91     + SALT +
92     "</td><td class='dataunderline'>ppm</td></tr><tr><td class='dataunderline'>Conductivity </td><td class
='dataunderline'>"
93     + COND +
94     "</td><td class='dataunderline'> millisiemens</td></tr><tr><td class='dataunderline'>Turbidity </td><td
class='dataunderline'>"
95     + TUR +
96     "</td><td class='dataunderline'> inches</td></tr><tr><td class='dataunderline'>Clarity </td><td class='d
ataunderline'>"
97     + CLAR +
98     "</td><td class='dataunderline'>inches</td></tr><table></span>";
99     //END "html" variable set-up
100

```

```

101 //set the location of the marker to be displayed on the map
102 var icon = "http://nia.ecsu.edu/reuomps2014/teams/water/teamimages/markers/PQTmarkers" + PT + ".
png";
103
104 //Load variable "marker" with constructor to create location on map
105 var marker = new google.maps.Marker({
106     map: map,
107     position : POINT,
108     icon : icon //.icon
109 });
110 bindInfoWindow(marker, map, infoWindow, html);
111 }
112 });
113 }
114
115 //Function to allow the user to select the marker and view the information balloon
116 function bindInfoWindow(marker, map, infoWindow, html) {
117     google.maps.event.addListener (marker, 'click', function() {
118         infoWindow.setContent (html);
119         infoWindow.open(map, marker);
120     });
121 }
122
123 //Function for retrieval of XML data from the php file
124 function downloadUrl(url, callback) {
125     var request = window.ActiveXObject ?
126         new ActiveXObject("Microsoft.XMLHTTP") :
127         new XMLHttpRequest;
128
129     request.onreadystatechange = function() {
130         if (request.readyState == 4) {
131             request.onreadystatechange = doNothing;
132             callback (request, request.status);
133         }
134     };
135
136     request.open('GET', url, true);
137     request.send(null);
138 }
139
140 function doNothing() {}
141
142 //]]>
143 </script>
144
145 <link href= "googledom.css" rel="stylesheet" type="text/css" >
146 </head>
147
148 <body onLoad= "load()" >
149 <table width= "1024" border= "1" align="center" cellpadding= "0" cellspacing= "0" >
150 <tr>
151     <td><table width= "100%" border= "0" cellspacing= "0" cellpadding= "4" >
152         <tr>
153             <td><strong >2015 Multimedia Team </strong><br>
154             Implementation of an interactive database interface utilizing HTML, JavaScript, PHP, and MySQL
in support of water quality assessments in the Northeastern North Carolina Pasquotank Watershed <b

```



```

r>      <a href="http://nia.ecsu.edu/ur/1415/teams/db/"   >BACK TO SITE </a></td>
155  </tr>
156  </table></td>
157  </tr>
158  <tr>
159  <td><table width= "100%" border= "0" cellpadding= "4" cellspacing= "0" class= "BGmenuTable" >
160  <tr>
161  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/17/11&SRCNAME=Knobbs%
20Creek" class= "main_menu" >Knobbs Creek - 6/17/11 </a></td>
162  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/6/11&SRCNAME=Sawyers%
20Creek" class= "main_menu" >Sawyers Creek - 6/6/11 </a></td>
163  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/15/11&SRCNAME=Pasquota
nk River" class= "main_menu" >Pasquotank River - 6/15/11 </a></td>
164  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/14/11&SRCNAME=Areneuse
%20Creek" class= "main_menu" >Areneuse Creek - 6/14/11 </a></td>
165  </tr>
166  <tr>
167  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/14/11&SRCNAME=Mill%20D
am%20Creek" class= "main_menu" >Mill Dam Creek - 6/14/11 </a></td>
168  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/30/11&SRCNAME=Newbegu
n%20Creek" class= "main_menu" >Newbegun Creek - 6/30/11 </a></td>
169  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/7/11&SRCNAME=Newbegun
%20Creek" class= "main_menu" >Newbegun Creek - 6/7/11 </a></td>
170  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/13/11&SRCNAME=Newbegu
n Creek" class= "main_menu" >Newbegun Creek - 6/13/11 </a></td>
171  </tr>
172  <tr>
173  <td>&nbsp;</td>
174  <td>&nbsp;</td>
175  <td>&nbsp;</td>
176  <td>&nbsp;</td>
177  </tr>
178  <tr>
179  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/26/13&SRCNAME=Knobbs%20Cr
eek" class="main_menu">Knobbs Creek - 6/26/13</a></td>
180  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/27/13&SRCNAME=Sawyers%20C
reek" class="main_menu">Sawyers Creek - 6/27/13</a></td>
181  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/24/13&SRCNAME=Pasquotank%2
0River" class="main_menu">Pasquotank River - 6/24/13</a></td>
182  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/21/13&SRCNAME=Areneuse%20
Creek" class="main_menu">Areneuse Creek - 6/21/13</a></td>
183  </tr>
184  <tr>
185  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/13/13&SRCNAME=Sawyers%20C
reek" class="main_menu">Sawyers Creek - 6/13/13</a></td>
186  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/21/13&SRCNAME=Mill%20Dam%
20Creek" class="main_menu">Mill Dam Creek - 6/21/13</a></td>
187  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/24/13&SRCNAME=Newbegun%20
Creek" class="main_menu">Newbegun Creek - 6/24/13</a></td>
188  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/20/13&SRCNAME=Knobbs%20Cr
eek" class="main_menu">Knobbs Creek - 6/20/13</a></td>
189  </tr>
190  <tr>
191  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/19/13&SRCNAME=Newbegun%20
Creek" class="main_menu">Newbegun Creek - 6/19/13</a></td>
192  <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/18/13&SRCNAME=Pasquotank%2

```

```

0River" class="main_menu">Pasquotank River - 6/18/13</a></td>
193 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/14/13&SRCNAME=Areneuse%20
Creek" class="main_menu">Areneuse Creek - 6/14/13</a></td>
194 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/14/13&SRCNAME=Mill%20Dam%
20Creek" class="main_menu">Mill Dam Creek - 6/14/13</a></td>
195 </tr>
196 <tr>
197 <td>&nbsp;</td>
198 <td>&nbsp;</td>
199 <td>&nbsp;</td>
200 <td>&nbsp;</td>
201 </tr>
202 <tr>
203 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/17/14&SRCNAME=Knobbs Creek"
class="main_menu">Knobbs Creek - 6/17/14</a></td>
204 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/11/14&SRCNAME=Sawyers%20C
reek" class="main_menu">Sawyers Creek - 6/11/14</a></td>
205 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/20/14&SRCNAME=Pasquotank Ri
ver" class="main_menu">Pasquotank River - 6/20/14</a></td>
206 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/10/14&SRCNAME=Areneuse%20
Creek" class="main_menu">Areneuse Creek - 6/10/14</a></td>
207 </tr>
208 <tr>
209 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/10/14&SRCNAME=Mill Dam Cree
k" class="main_menu">Mill Dam Creek - 6/10/14</a></td>
210 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/20/14&SRCNAME=Newbegun%20
Creek" class="main_menu">Newbegun Creek - 6/20/14</a></td>
211 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/9/14&SRCNAME=Knobbs%20Cre
ek" class="main_menu">Knobbs Creek - 6/9/14</a></td>
212 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/6/14&SRCNAME=Sawyers%20Cr
eek" class="main_menu">Sawyers Creek - 6/6/14</a></td>
213 </tr>
214 <tr>
215 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/4/14&SRCNAME=Pasquotank%20
River" class="main_menu">Pasquotank River - 6/4/14</a></td>
216 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/4/14&SRCNAME=Areneuse%20Cr
eek" class="main_menu">Areneuse Creek - 6/4/14</a></td>
217 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/2/14&SRCNAME=Mill%20Dam%2
0Creek" class="main_menu">Mill Dam Creek - 6/2/14</a></td>
218 <td><a href="http://nia.ecsu.edu/googleDOM_example.html?DATE=6/4/14&SRCNAME=Newbegun%20
Creek" class="main_menu">Newbegun Creek - 6/4/14</a></td>
219 </tr>
220 </table></td>
221 </tr>
222 <tr>
223 <td>&nbsp;</td>
224 </tr>
225 <tr>
226 <td> <div id="map" style="width: 1024px; height: 900px"></div></td>
227 </tr>
228 <tr>
229 <td>&nbsp;</td>
230 </tr>
231 </table>
232
233

```

```
234 </body>
235
236 </html>
```