

Interactive World Data
By Patrick Tsai
Git repo: pattsai/slp

1. Objectives. Describe the problem being solved and specify the extent to which that problem will be solved by this project.

Have you ever watched the news and asked yourself, “Why did the government make that decision?” There is usually a relationship between data and decision making. The general public has access to government agency data. The problem is that much of the data released is indecipherable, raw data contained within stacks of spreadsheets. Some people may know how to translate the data, the rest of us are reliant on someone else’s “take”, usually promoting a political agenda. Although the general public may not be experts, we do deserve a place to interact with raw data, detached from political slant, and presented in an easily accessible/intelligible manner. Additionally, researchers and decision makers should also have access to this raw data for all the same reasons as the general public. However, decision makers should not use access to raw data as an alternative to unbiased expert analysis.

My project, Interactive World Data, seeks to create a web based, interactive display of United Nations and World Bank data on Hunger, Poverty, Greenhouse Gas Emissions, Import, and Export data. Eventually, I’d like to include all official international government data categories within my database. My hope, not currently met, is to show the relationships between different categories of data (such as, how does increased exports affect GHG emissions or poverty). I want to present data in an easy-to-consume visual manner using graphs. My end goal is to allow users to project how data in these categories will change based on the relationship it has with another category. I was unable to complete the predictive part of my project.

Objectives:

- 1.) Gather data from trusted official sources
- 2.) Upload data to a database
- 3.) Create a development environment that allows
 - a.) Web access
 - b.) Webpage communication to a database
 - c.) Database queries
 - d.) Graphical display of data
- 4.) Allow user to choose data from a particular country
- 5.) Use database queries to display country data
- 6.) Display data graphically
- 7.) Allow user input for data manipulation
- 8.) Predictive algorithm applied to data

Central to my project is the belief in free and open access to unbiased data and information. I believe the creation of Interactive World Data (a free, open source, web based application) will empower everyone by allowing access to unbiased raw data in an easily consumable manner. And once the predictive modeling component is complete, Interactive World Data will give researchers and policy makers an invaluable tool to guide policy decisions.

2. Alternative Solutions. Describe results of your research. Discuss solution alternatives considered and justification for the selected solution.

My research has found a market for applications or outlets that display raw data in a graphical way. Two of the organizations I have looked into Our World Data and GapMinder, are free and web based. Our World Data and GapMinder focus on displaying social data. Tableau, the last application I critiqued, is web based with a pricing structure. Tableau appears to be used across industry and governments.

Each of these applications has the same end goal, to present data to the user in a useful and easily consumable manner. Interactive World Data has the same end goal. Our World Data has a carefully curated pool of data relationships it presents to the user. Similarly, GapMinder curates their data and presents well thought out and established relationships. Both OurWorld Data and GapMinder are the result of Data Analytic expert analysis of social data. The downside is that not all data categories are present and therefore there may be an absence of important data relationships. My current project proposal also suffers from a lack of data categories. In the future I hope to include all official international government data categories.

Tableau offers a full suite of data analytic and display tools. Tableau also offers users the ability to import data. Essentially, I would like to make an open source Tableau application using set data from every internationally recognized data category.

Foundationally, my project provides free access to unbiased data and analysis, and that idea directs development decisions. This foundational belief is why I have chosen the approach outlined in my objective and why I did not choose a similar approach as the 3 organizations described above.

3.Solution Specifications. Provide complete solution specifications, including any of the following items that apply:◦System diagram (input and output definition)◦Block diagram (Major component of design)◦High level design◦Code and schematics (code can be posted on GIT and a link provided)

1. Gathering data from trusted official sources
 - a. Annual data was gathered from 2000 to 2018
 - b. Import and Export data was gathered from the World Bank's WITS (world integrated Trade Solution) database using the UN COMTRADE advanced query portal.
(<http://wits.worldbank.org/WITS/WITS/AdvanceQuery/BulkExport/BulkExportQueryDefination.aspx?Page=BulkExport>)
 - c. Data for Poverty, Hunger, and GHG emission was downloaded from the United Nations Sustainable Development GoalsIndicators Database.
 - i. Poverty - Goal 1 Target 1.1 By 2030, eradicate extreme poverty for all people everywhere, currently measured as people living on less than \$1.25 a day
 - ii. Hunger - Goal 2 Target 2.1.1 Prevalence of undernourishment (%)
 - d. Data for GHG emissions was downloaded from the United Nations Framework Convention on Climate Change.
 - i. Annex 1 Time Series (https://di.unfccc.int/time_series)
2. Identify a computer with the resources to build the application
 - a. The Hardware I used was an old Lenovo X240 Thinkpad
 - i. Intel Core i5-4300U CPU
 - ii. 64-bit OS
 - iii. Updated with 8GB RAM
 - b. I installed Linux based Ubuntu 18.04.4 LTS
 - c. I installed MySQL Database
 - i. Installed mysql-server
 - ii. Installed mysql-client *** `sudo apt-get install libmysqlclient-dev`
3. Create a development environment that allows
In addition to Installing Ubuntu and MySQL
 - a. Installed Apache 2 as a webserver to host files
HTML files are served from /var/www/html/ folder
 - i. `sudo apt-get install apache2`

b. Installed PHP Version 7.4.6

PHP files are served from /usr/lib/cgi-bin/

- i. `sudo apt-get install php libapache2-mod-php`

c. Webpage communication with database

- i. Installed CURL

1. `sudo apt-get install curl`

d. Database queries

- i. Initially I was using C programming language to write my database queries, however outputting to html had to be wrapped in a print statement. This became cumbersome (see index.c and main.c in git repo).

- ii. I later chose PHP - PHP Hypertext Preprocessor language to write the database queries.

1. I found writing database query code integrated with HTML code simpler using PHP

- iii. Using PHP (or C) to query the MySQL database is dependent on the Common Gateway Interface, and I had to enable the cgi-bin to serve my PHP files from

Configure CGI

I looked around the `/etc/apache2` directory, which is the standard place to find the configuration files of Apache. I've found the `/etc/apache2/conf-available/serve-cgi-bin.conf` file that has a symbolic link from `/etc/apache2/conf-enabled/serve-cgi-bin.conf`. It has a section that maps the `/cgi-bin` path in the URLs to the `/usr/lib/cgi-bin/` directory in the hard disk and enables CGI execution in this directory.

examples/apache/serve-cgi-bin.conf

```
1. <IfModule mod_alias.c>
2.     <IfModule mod_cgi.c>
3.         Define ENABLE_USR_LIB_CGI_BIN
4.     </IfModule>
5.
6.     <IfModule mod_cgid.c>
7.         Define ENABLE_USR_LIB_CGI_BIN
8.     </IfModule>
9.
10.    <IfDefine ENABLE_USR_LIB_CGI_BIN>
11.        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
12.        <Directory "/usr/lib/cgi-bin">
13.            AllowOverride None
14.            Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
15.            Require all granted
16.        </Directory>
17.    </IfDefine>
18. </IfModule>
19.
20. # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

That's not enough though. We also need to enable the CGI module of Apache.

The installed modules of Apache can be found in the `/etc/apache2/mods-available` directory. The cgi module is called `cgi.load`

The enabled modules have symbolic links in `/etc/apache2/mods-enabled`, but as I found out, the CGI module did not have a symbolic link there: The CGI module was not enabled by default.

```
$ cd /etc/apache2/mods-enabled
$ sudo ln -s ../mods-available/cgi.load
```

Added the symbolic link.

- iv. Additionally, I installed phpMyAdmin. phpMyAdmin provides a graphical user interface for viewing MySQL server Databases and tables. Users are able to run queries as well within the application.

e. Graphical display of data

I originally installed and enabled the graphical display elements after creating my database and trying to serve both C and PHP files, since I didn't have the knowledge of how to make it all work. After struggling without a clear plan, it's apparent to me that the graphical display element installation should be completed while setting up the server.

- i. Install Node.js - javascript package

Javascript files are served from /usr/share/javascript folder

1. `curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash - sudo apt-get install -y nodejs`

- ii. JSON support comes pre-compiled with php.
you can check for installed php extensions with:
`sudo dpkg --get-architecture | grep php | grep -i <extension_name>`
 - iii. Install Chart.js
Chart.js is an open source library that uses javascript. Chart.js builds charts from data passed in json format.
 - 1. `npm install chart.js --save`
4. Upload data to a database
- a. Before uploading all the data needed to be placed in individual comma separated spreadsheets based on country. I used Excel to create the .csv spreadsheets for 196 countries in the United Nations. Each country spreadsheet contains the headers imports, exports, hunger, poverty, ghg_emissions.
 - b. Spreadsheets were named with the 3-letter UN abbreviation for each country.
unzip, rename, and alter files -
 - i. Changed the file names of all the country data spreadsheets (useful changes):
 - Used the rename and sed command: `rename 's/^\.*-/' *.csv` (removes everything up to the "-" in the file name)
 - Changed the column names to lower case using the sed command `sed -i 's/GHG Emissions/ghg_emissions/g' *.csv`
 - Changed Excel line breaks (cntrl V, cntrl M) into \n `sed -i 's/\n/g' *.csv`
 - c. Created a MySQL Database named countrydata
 - d. Wrote a Bash script to create a table based on each spreadsheet, named with the 3-letter abbreviation for each country. [With regard to my entire project, I am most proud of this script. It can be found in my slp repo file name: `mysqlUpload.sh`]

```
#!/bin/bash
dir="country-data"
for entry in "$dir"/*
do
    echo "$entry"
    inputString=$entry
    subString=$(echo $inputString|cut -b 14-16)
    echo $subString

    sudo mysql -u "root" -e "USE countrydata; SELECT DATABASE(); CREATE TABLE $subString(year INT(4), country VARCHAR(128), code VARCHAR(4), export FLOAT(14,2), import FLOAT(14,2), hunger FLOAT(7,2), poverty FLOAT(7,2), ghg_emissions FLOAT(14,2));"
    sudo mysql -u "root" -e "USE countrydata; LOAD DATA LOCAL INFILE './$entry' INTO TABLE $subString FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 ROWS;"
done
```

Explanation of Script

- i. The Bash script first assigns the folder to look at (dir=)
- ii. Create a for loop to look at every file in the folder
- iii. Entry var will have the folder name that the csv files are in, therefore the script "cuts" the directory name off and leaves off the .csv at the end (hence, -b 14-16) since all files are 3 letters this command works well. The script then uses the resulting string to name a table within database
 - 1. Create the table with subString
 - 2. Assign type to each table column
- iv. Then upload the data
 - 1. Ignoring the first row
- v. The script increments through an entire folder until every file has been uploaded
- vi. The script then ends

5. Allow user to choose data from a particular country

Once the development environment was created, I started writing my code in HTML/PHP/MySQL (originally I used C but found it too cumbersome).

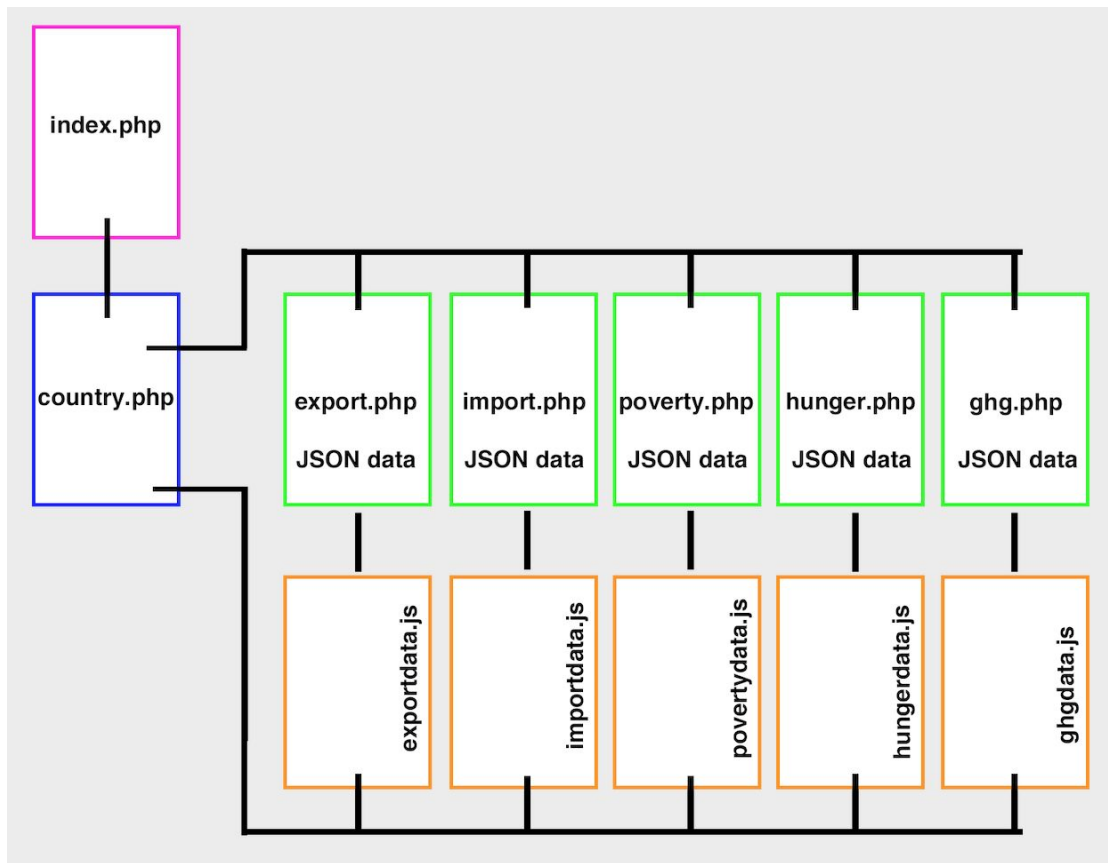
- a. User country choice was created with hypertext links added to the country names, using HTML.
 - i. A database query is used to pull all countries within the database and list them, alphabetically using the country name found in the country column of each countries table.
 - ii. Each country listed has an assigned country code added to the end of the hyperlink (i.e url?code=USA) for identification on the generic country page, which displays country data.
** side note: Learning how to pass data through a URL made the project worth it.
 - b. I added a dropdown tab which displays all the "linked" countries on the home page.
 - i. The tab was created with CSS
6. Use database queries to display country data
 - a. Using PHP's mysqli functionality, queries are established to the countrydata database.

```
<?php
echo "<p>" . $countryName . " - " . $codeName . "</p>";

if (isset($_GET["code"])) {
    $showAll = $mysqli->query("SELECT * FROM ". $codeName . " ");
    if ($showAll->num_rows > 0) {
        echo "<table id='countryTable'>
            <tr>
                <th class='year'>YEAR</th>
                <th class='export'>EXPORTS</th>
                <th class='import'>IMPORTS</th>
                <th class='poverty'>POVERTY</th>
                <th class='hunger'>HUNGER</th>
                <th class='ghgemissions'>GHG EMISSIONS</th>
            </tr>";
        while ($row4 = $showAll->fetch_assoc()) {
            echo "<tr>
                <td>" . $row4["year"] . "</td>
                <td>" . $row4["export"] . "</td>
                <td>" . $row4["import"] . "</td>
                <td>" . $row4["poverty"] . "</td>
                <td>" . $row4["hunger"] . "</td>
                <td>" . $row4["ghg_emissions"] . "</td>
            </tr>";
        }
        echo "</table>";
    }
}
?>
```

7. Display data graphically

Diagram of data movement through webpages and graphical display. Starting when a user enters the index page and ending with the graphical display on the country.php page.



8. Allow user input for data manipulation
 - a. I was unable to complete
9. Predictive algorithm applied to data
 - a. I was unable to complete

4.Verification. Test plan and test results verifying that the project objectives have been met.

The majority of my project was creating a development environment. Creating the development environment allowed me to create pages accessible from a website that query the database, return and display results graphically.

Testing Plan:

- 1.) Write database query using PHP mysqli
- 2.) Write an accessible webpage using the Common Gateway Interface
- 3.) Use browser to view the output of the database query in a table format
- 4.) Push integer results in JSON format to another php page (category_header.php)
- 5.) Use javascript to pull results and render a graphical representation of data
- 6.) Call from javascript from the php page rendering country data

Verifying results:

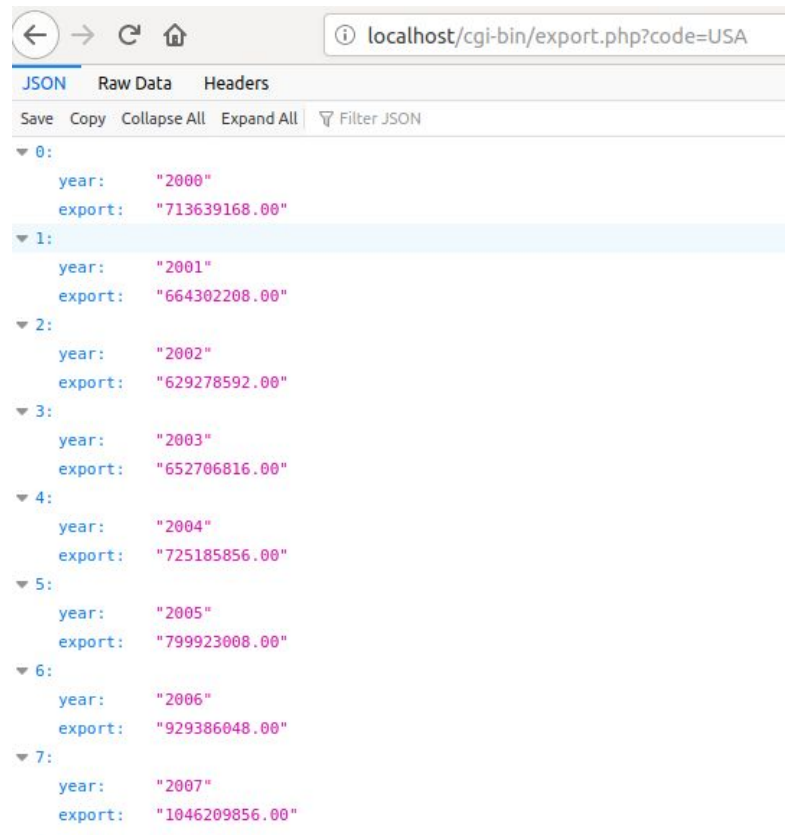
- 1.) Verification that the webserver is successfully displaying the default webpage with a listing of all the countries
 - a.) Using a browser I attempted to access <http://localhost/cgi-bin/index.php>
 - b.) Upon page load I selected the countries tab, and made sure it listed all of the countries within the database.

- 2.) Once a country is selected from the country list page, data specific to the country is displayed in a table format on the country page.
 - a.) I was able to verify the query was correct by comparing values between query output and the database. Additionally, I compared the query output with the spreadsheet originally uploaded to the database
- 3.) Graphical display verification - two part verification

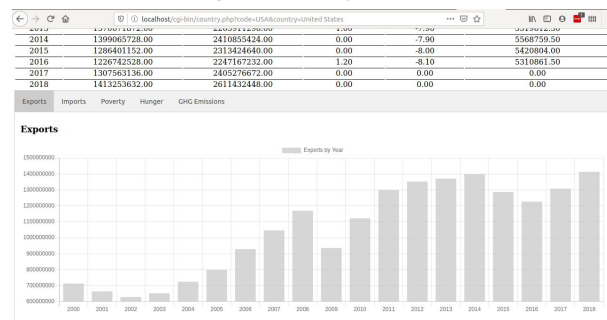
I struggled a lot with graphing.

Graphing is dependent on:

 - a.) Running a database query that outputs data in JSON format (i.e seen in file export.php)
 - i.) Verification seen http://localhost/cgi-bin/export.php?code=country_code
i.e <http://localhost/cgi-bin/export.php?code=USA>



- b.) Next a javascript file (i.e seen in file exportdata.js) reads the JSON formatted data
- c.) Then renders a bar graph using the Chart.js package on the individual country page
 - i.) Verification of the b-c is seen by the rendered graph on the country page
i.e <http://localhost/cgi-bin/country.php?code=USA&country=United%20States>



5.Future Enhancements. Describe your recommendation for potential enhancements to your solution.

I would like to make two short term enhancements, and one long term overhaul.

Short term:

- 1.) I would like to include more categories. I would like to include every UN Sustainable Development Goal Category, found at <https://unstats.un.org/sdgs/indicators/database/>
- 2.) I would like to create a form for users to manipulate data, and see how it affects other categories of data.
 - a.) The functionality would include linear regression analysis, and would use the line equation to project how one category affects the another.

Long term:

- 1.) My long term goal is to expand the data projection model. I would like to take into account all of the categories at once using a linear fit model. At the moment I'm taking CSE280, which has provided insight in machine learning algorithms. In order to expand on my long term goal, I believe I would have to switch to using the python programming language, and redevelop my development environment. Creating a new development environment may be easier due to using built in graphing with pyplot.

6.Lessons Learned. How would you improve your development and design process based on your experience and learning through this project.

I failed so many times. I learned so much.

My major lessons:

- 1.) Installing and getting communication between Linux, Apache, PHP, MySQL, Node.js, Chart.js, JSON.
 - a.) The whole process of creating a working development environment took the most time. I became comfortable installing packages with linux.
- 2.) My major issues revolved around javascript and JSON. Undoubtedly, the hardships stem from my lack of knowledge. Many times my output wasn't in JSON specific format. Additionally, there are a number of attributes you must assign in Javascript to read the data. I really didn't know what I needed to configure and what was extraneous.
- 3.) Additionally, my use of PHP and MySQL was simple. This was also due to my lack of knowledge. Now that I'm getting familiar with object oriented programming, I believe I could better use PHP and classes to render data from MySQL queries much cleaner.

I would improve development first by installing a prebuilt LAMP stack. Creating this environment on my own was too time intensive. I'm glad I got to practice doing it myself, however, it took too much time away from actual web development. Lastly, I would take a class on Javascript and JSON in order to understand the errors I made rendering data.

7.License Terms. You may make these more restrictive, as you see fit. Here are default license terms for an open-access project:

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.