PHP / MySQL Database Project – “Top Apps”

Andrew Wong || 12SOF1 2017 || [github.com/bearbear12345/school\_2017\_12SDD](https://github.com/bearbear12345/school_2017_12SDD)

# Project Overview

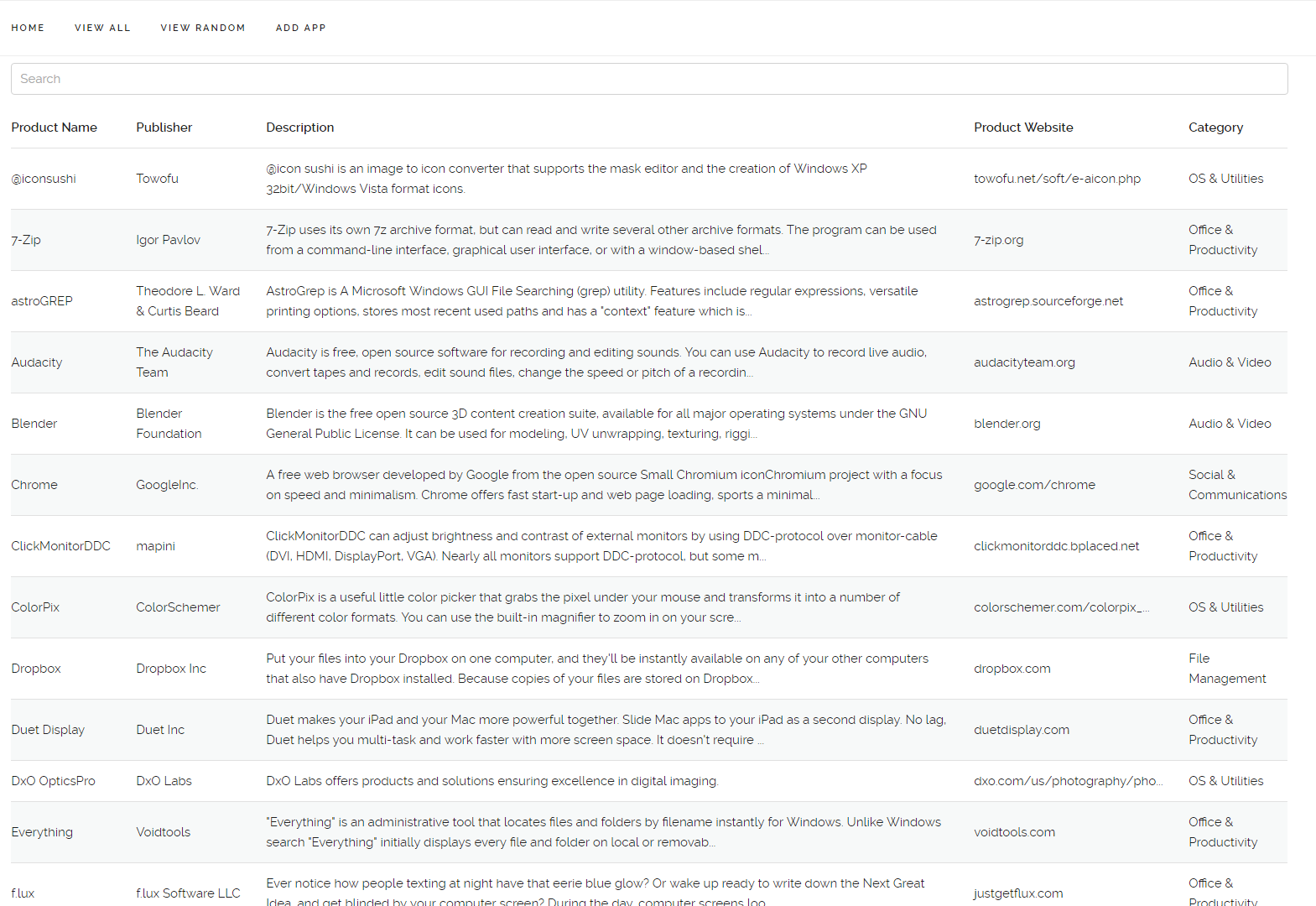
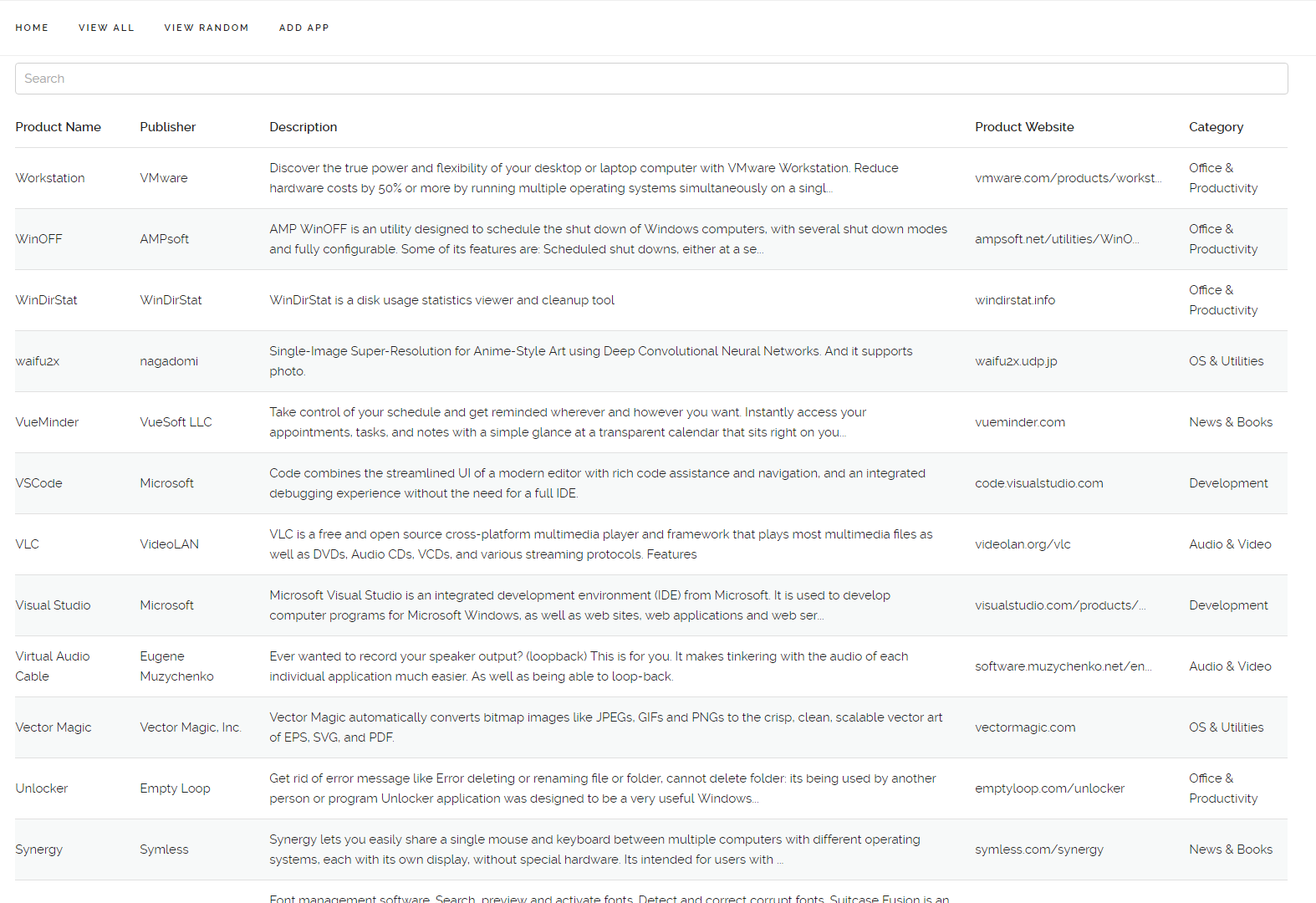
Having been asked countless times by many people about what software I use and recommend, I used this idea as the theme of my project. The database webpage shows a list of software that I use, providing relevant information (software publisher, a description of it, and a link to its product website page) about the application.

If the user clicks on an application listing, they will be brought to a page with more details about that application.  
The user is able to search for a specific item, as well as filter the search results.  
They are also able to add and modify the application details

# Website Features

## | Sorting

By clicking on either the ***Product Name****,* ***Publisher***, or ***Category*** table headers, the webpage will show a sorted list of items, going either ascending or descending in name.

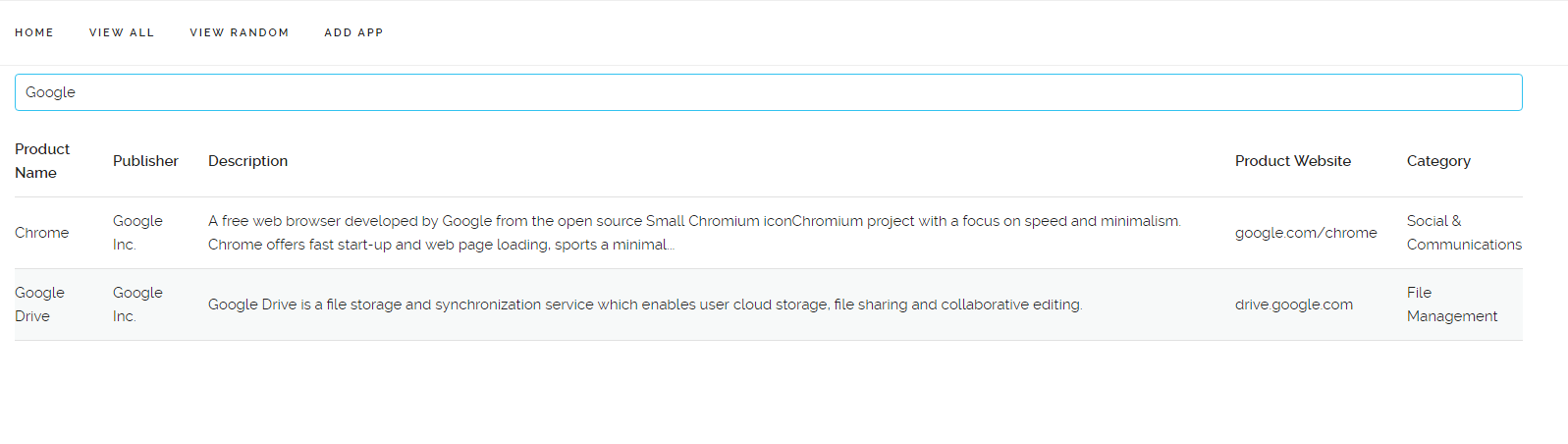
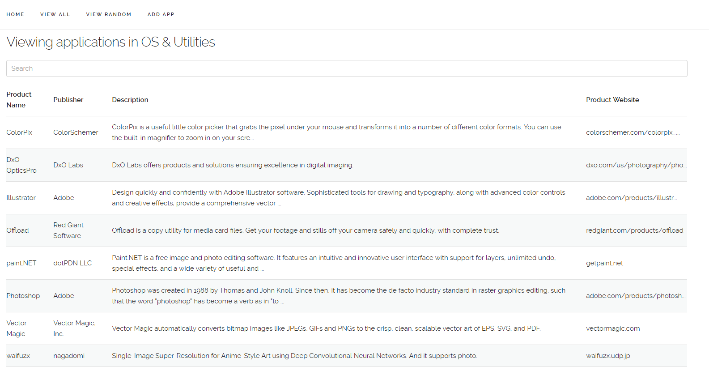
 

Sort by Product Name, A to Z

Sort by Product Name, Z to A

## | Search Filtering

Application listings can be filtered by name or category. This can be done respectively by typing in the Search bar, or selecting a category from the rightmost table column.

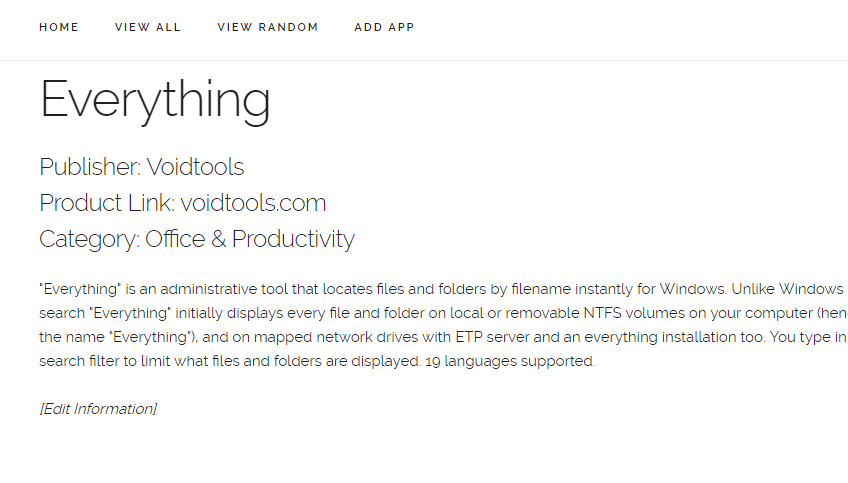
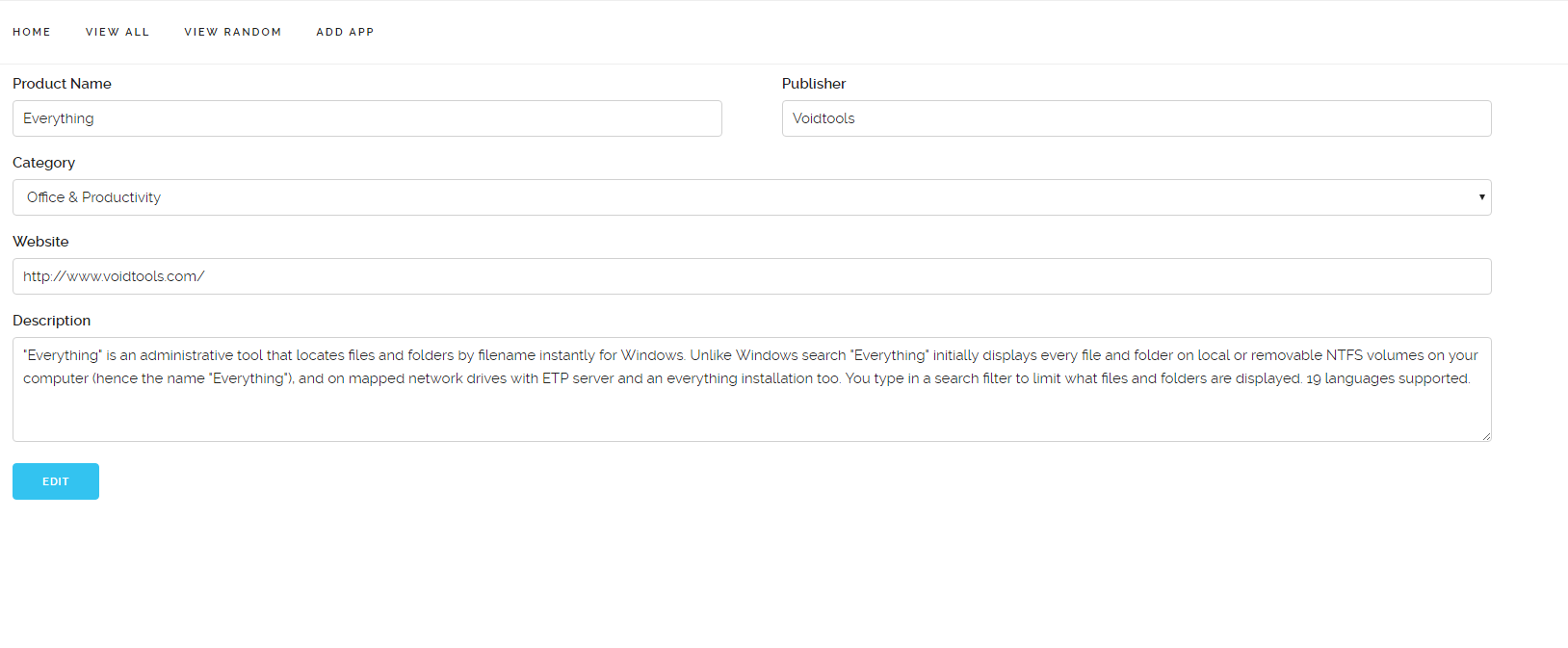
 

Filtering by category “OS & Utilities”

Searching for “Google”

## | Application Information Page

By clicking on an application’s table row, the user will be directed to that application’s page, where more information of the application is shown. Application details can be edited by clicking on the ***[Edit Information]*** link following the application description.

Application Information Editing Page

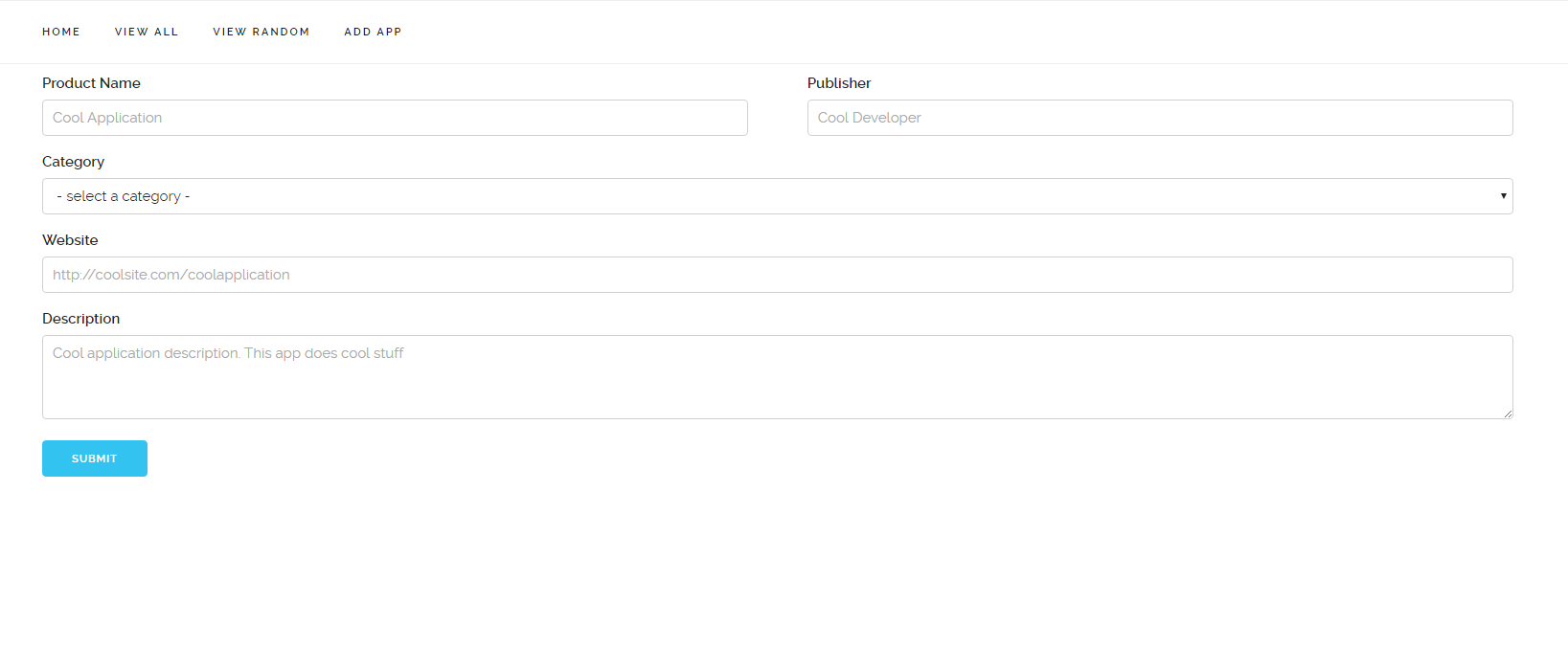
Application Information Page

## | Random Application

By clicking on the ***View Random*** link the user will be directed to a random application information page.

## | Adding Applications

By clicking on the ***Add App*** link the user will be able to add in new application details into the database.



## | Scroll To Top

A ‘back to top’ function is included to scroll to the top of the page (hint: turn on sound!)

# Database Structure

## | Database Table Overview

My database consists of two tables, applicationdb and categorydb. The former table contains records about each application in the database (name, publisher, description, website link, *categoryid*). The latter contains a list of category names, associated to a unique ID (referred to as *categoryid*).

## | Data Dictionary

### Table: applicationdb

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| AID | Integer (primary key) | ID number of application entry |
| Product Name | String | Name of the application |
| Publisher | String | Application publisher |
| Description | String | Information about the application |
| Product Link | String | Website to the product page |
| CategoryID | Integer | Category ID associated to the product |

### Table: categorydb

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| CategoryID | Integer (primary key) | ID number of category entry |
| CategoryName | String | Name of the category |

## | Data Redundancy & Linking Tables

Data redundancy was reduced by creating a category table, and associating each application with an ID linked to a category. Effectively, this would reduce the size of the database (as the actual name of the category is only stored once, compared to being stored in each record), and also allowed for easier searching and filtering functions. I decided against also doing the same with the Publisher field, as there were more distinct entries than repeated ones for this number of listings, however this would be advisable for a larger database.

# Inspecting the database source code (KEY SECTIONS)

### | Initialising Database Connections: ./lib/dbconnect.php

<?php

$db\_server = "localhost"; *// Connect to server at localhost:3306*

$db\_username = "andreww"; *// Login as user "andreww"*

$db\_password = "andrew"; *// Authenticate with password "andrew"*

$db\_database = "andreww"; *// Open database "andreww"*

$db = mysql\_connect($db\_server, $db\_username, $db\_password);

mysql\_select\_db($db\_database)

?>

This php file connects to the MySQL server, and opens the database for the other functions to use.

### | Editing and Adding Application Details: ./manage.php

**include\_once** "lib/dbconnect.php";

*/\* Handle form submits by adding or editing the data in the SQL database \*/*

**if** ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

...

**if** ($aid) {

*// if $aid is present, then the data is for an application edit*

$sql = "UPDATE `applicationdb` SET `Product Name` = '**$inputName**', `Publisher` = '**$inputPublisher**', `Description` = '**$inputDescription**', `Product Link` = '**$inputWebsite**', `Category` = **$inputCategory** WHERE `aid` = **$aid**";

} **else** {

*// if $aid doesn't exist, then the data is for a new application*

$aid = mysql\_num\_rows(mysql\_query("SELECT \* FROM `applicationdb`", $db)) + 1; *// Get next application ID*

$sql = "INSERT INTO `applicationdb` (`AID`, `Product Name`, `Publisher`, `Description`, `Product Link`, `Category`) VALUES (**$aid**, '**$inputName**', '**$inputPublisher**', '**$inputDescription**', '**$inputWebsite**', **$inputCategory**)";

}

$s = mysql\_query($sql, $db); *// Execute query*

header("Location: query.php?aid=**$aid**"); *// Redirect to application page*

}

"UPDATE `applicationdb` SET `Product Name` = '**$inputName**', ... WHERE `aid` = **$aid**";  
This SQL statement edits a record in the database  
  
"INSERT INTO `applicationdb` (`AID`, ...) VALUES (**$aid**, ...)";  
This SQL statement adds a new record into the database

*/\* Get existing information for information editing \*/*

**if** (isset($\_GET['aid'])) {

...

$sql = "SELECT `Product Name`, `Publisher`, `Description`, `Product Link`, `CategoryID`, `CategoryName` FROM `applicationdb`, `categorydb` where applicationdb.Category = categorydb.categoryid AND applicationdb.aid = **$aid**";

$s = mysql\_query($sql, $db);

**if** (!mysql\_num\_rows($s)) header("Location: query.php"); *// Check that the application id exists, else redirect to the entire listing*

$record = mysql\_fetch\_array($s);

...

$sql = "SELECT `categoryid`, `categoryname` FROM `categorydb`";

$s = mysql\_query($sql, $db);

**while** ($record = @mysql\_fetch\_array($s)) {

**echo** "**\n** <option value='**{**$record['categoryid']**}**'" . (@$appcati == $record['categoryid'] ? " selected" : "") . ">**{**$record['categoryname']**}**</option>";

}

"SELECT ... FROM `applicationdb`, `categorydb` where applicationdb.Category = categorydb.categoryid AND applicationdb.aid = **$aid**";  
This SQL statement queries for a specific application entry in the database

### | Retrieving Application Details: ./query.php

...

$order = " ORDER BY `**$sortsubject**` **$sortdirection**";

...

$sql\_category = isset($\_GET['cat']) ? " AND applicationdb.Category = " . $\_GET['cat'] : ""; *// Set category filter if present*

$sql = "SELECT `AID`, `Product Name`, `Publisher`, `Description`, `Product Link`, `CategoryID`, `CategoryName` FROM `applicationdb`, `categorydb` where applicationdb.Category = categorydb.categoryid" . $sql\_category . $order;

$s = mysql\_query($sql, $db);

...

**while** ($record = @mysql\_fetch\_array($s)) {

...

**echo** " <tr onclick='window.location.search=**\"**aid=**$appiden\"**'><td class='searchName'>**$appname**</td><td class='searchPublisher'>**$apppubl**</td><td>**$\_appdesc**</td><td><a href='**$appsite**'>**$\_appsite**</a></td>" . (!isset($\_GET['cat']) ? "<td class='searchCategory'><a href='query.php?cat=**$appcati**'>**$appcate**</a></td>" : "") . "</tr>**\n**";

}

**echo** " </tbody>**\n** </table>**\n** <script src='js/site.js'></script>**\n**";

} **else** {

*/\* Specific application \*/*

$sql = "SELECT `Product Name`, `Publisher`, `Description`, `Product Link`, `CategoryID`, `CategoryName` FROM `applicationdb`, `categorydb` where applicationdb.Category = categorydb.categoryid AND applicationdb.aid = **$aid**";

$s = mysql\_query($sql, $db);

$record = mysql\_fetch\_array($s);

...

**echo** " <h1>**$appname**</h1>**\n** <h5>Publisher: **$apppubl**</br>**\n** Product Link: <a href='**$appsite**'>" . preg\_replace('/.\*?:\/\/(?:www.)?(.+?)\/?$/', '${1}', $appsite) . "</a></br>**\n** Category: <a href='./query.php?cat=**$appcati**'>**$appcate**</a></h5>**\n** **$appdesc**</br>**\n** </br>**\n** <i><a href='./manage.php?aid=**$aid**'>[Edit Information]</a></i>**\n**";

}

$order = " ORDER BY `**$sortsubject**` **$sortdirection**";  
This SQL statement snippet modifies the order of the database query results  
  
$sql\_category = isset($\_GET['cat']) ? " AND applicationdb.Category = " . $\_GET['cat'] : "";  
This SQL statement snippet specifies the categoryid to filter from the database if necessary  
  
"SELECT ... FROM `applicationdb`, `categorydb` where applicationdb.Category = categorydb.categoryid" . $sql\_category . $order;  
This SQL statement queries for all application entries in the database optionally filtered by category, and optionally returned in a sorted order

### | Viewing a Random Application: ./random.php

<?php **include\_once** "lib/dbconnect.php";

$sql = "SELECT \* FROM `applicationdb`";

$s = mysql\_query($sql, $db);

$num\_rows = mysql\_num\_rows($s);

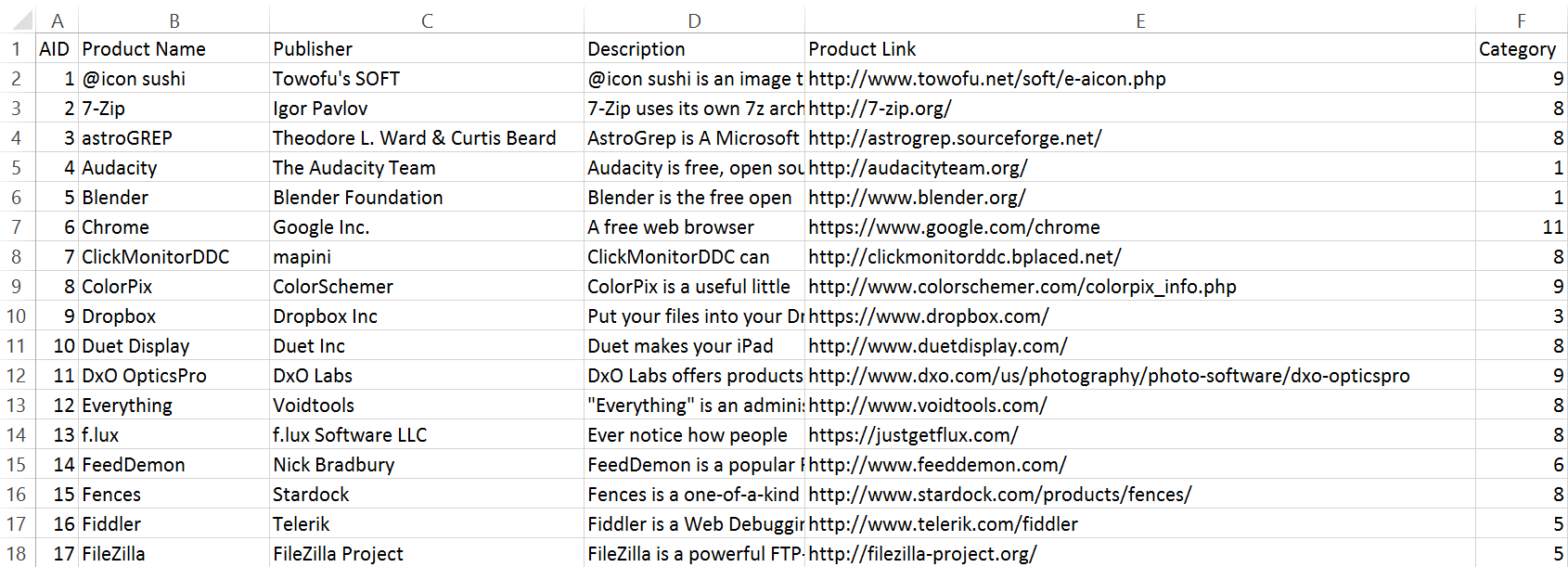
header("Location: query.php?aid=" . mt\_rand(1, $num\_rows));

?>  
This php file opens the applicationdb table, and counts the number of rows (that is, the number of application entries) in the database, and redirects the user to a random application page

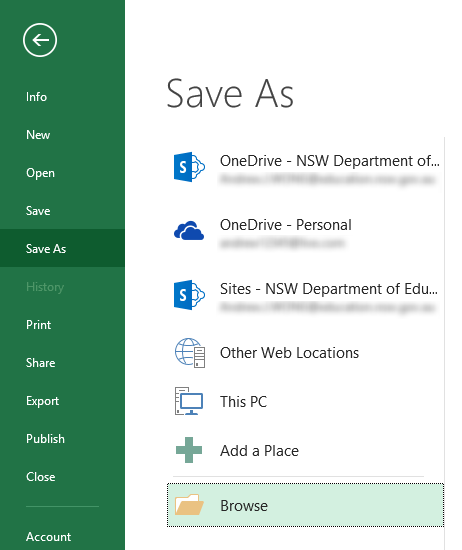
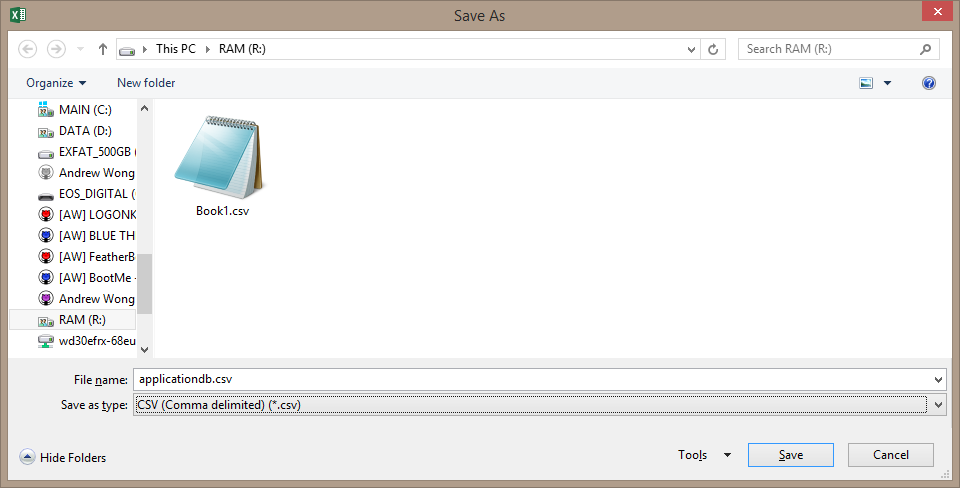
# Creating a MySQL Database with phpMyAdmin

*phpMyAdmin is an interactive website tool to help assist in the administration of MySQL databases*  
Designing and populating a database can all be done internally inside phpMyAdmin, however for simplicity we will be using Microsoft Excel to enter in data, which can then later be imported into the database

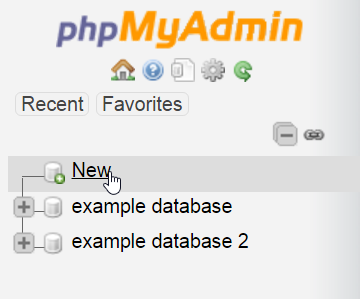
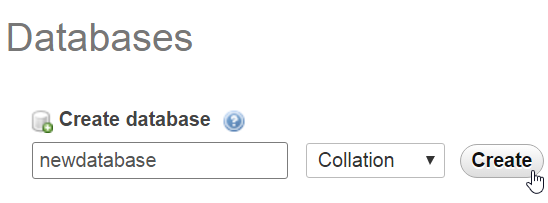
1. Create an excel spreadsheet with your data (optionally including headers)

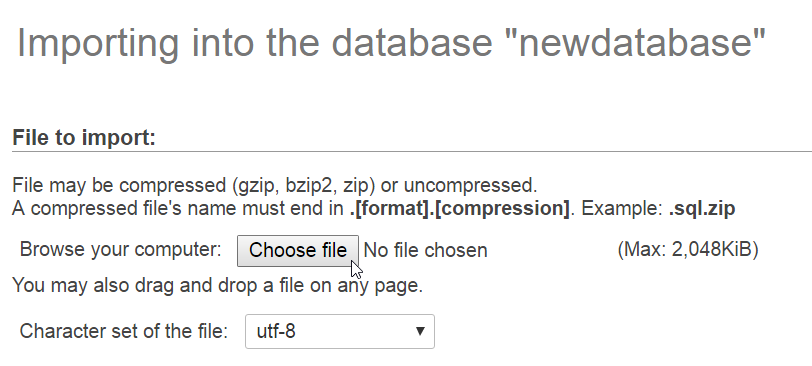


1. Export your data as a CSV file.

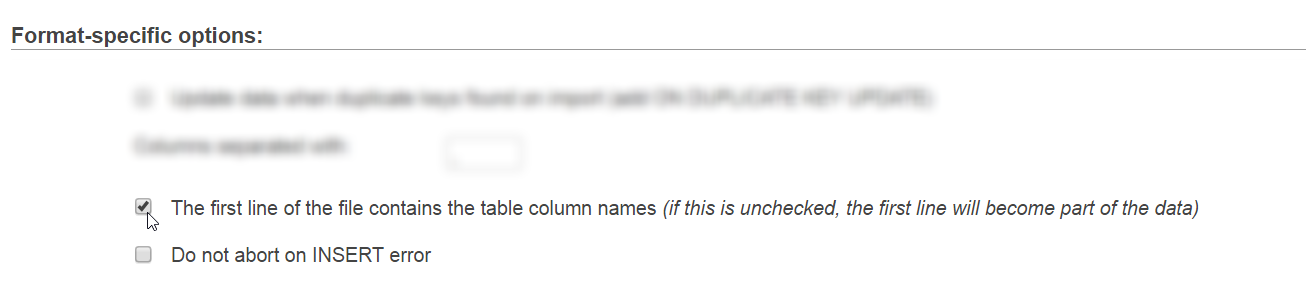
 

1. Navigate to phpMyAdmin with your web browser
2. Create a new database

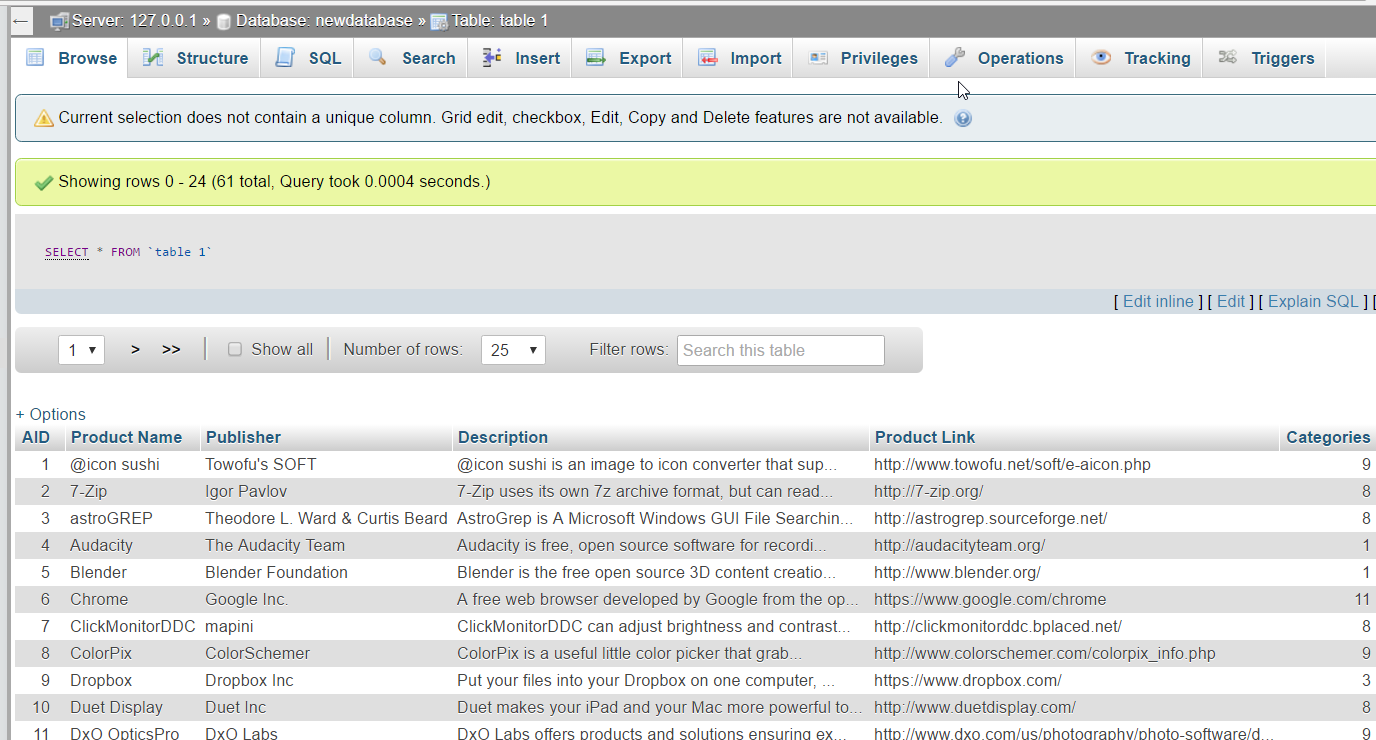
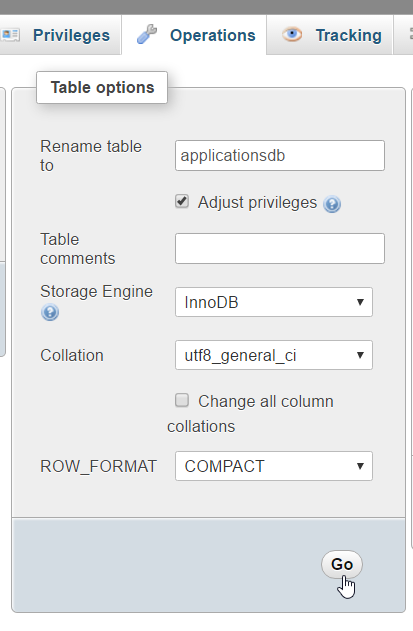
 

1. Import the CSV files into the new database  
   

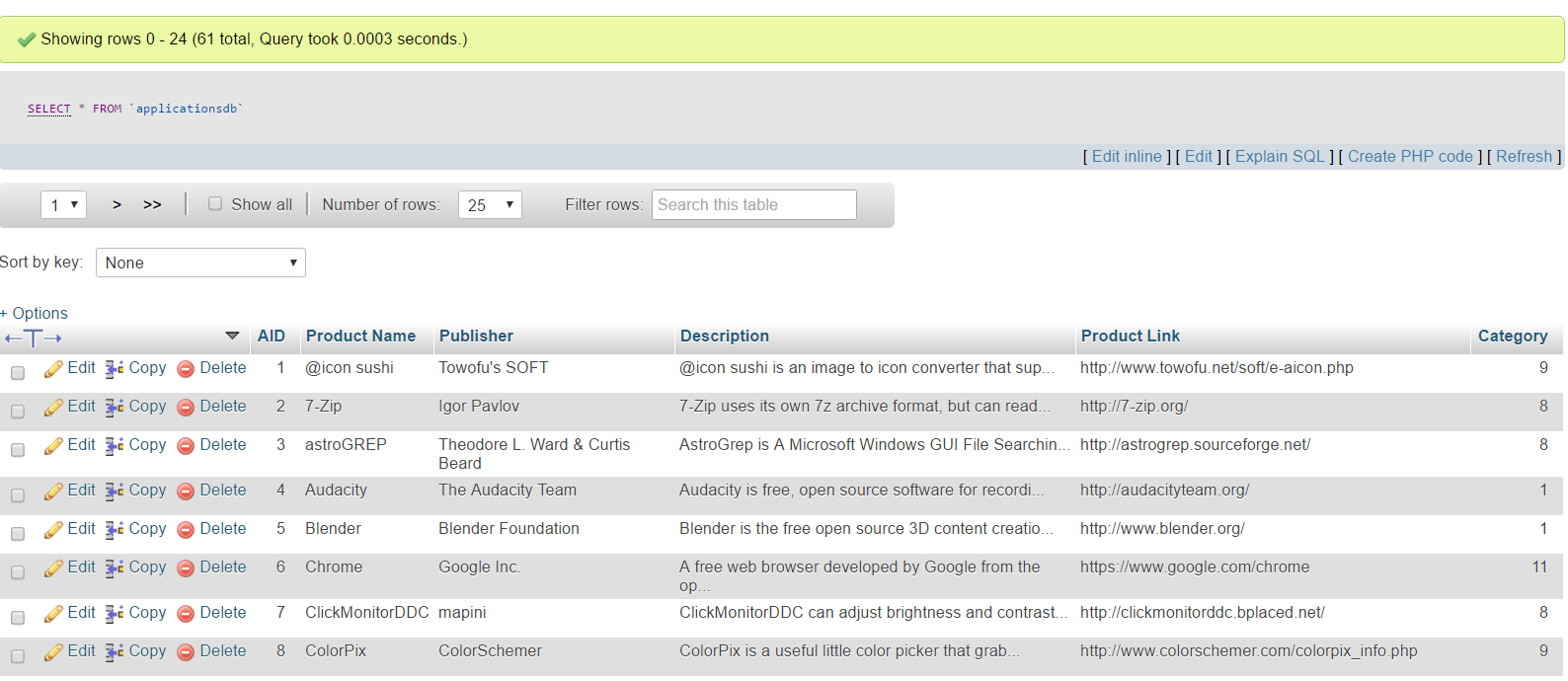
5.5) If the data was created with headers, select the following option



1. Rename the table

**Done!**



# More Source Inspection (Website specifics)

### | Header, Footer, Includes: ./globals/header.php & ./globals/footer.php

*// START OPTIONS*$pageElevator = **false**;  
$pageNavigation = **true**;  
$pageTitle = "TopApps";  
*// END OPTIONS*

All php webpages in this project include a variation of the above. These variables set options for the header.php file.

**include\_once** 'globals/header.php';

This line executes the header.php file located in the globals directory

**include\_once** 'globals/footer.php';

This line executes the footer.php file located in the globals directory

File: header.php

**if** (@$pageElevator) {**echo** "<script src='js/elevator.min.js'></script>**\n**";}

If the $pageElevator variable is set to true, insert the Elevator JS page scrolling JavaScript library

**echo** "<title>**$pageTitle**</title>**\n**";

Insert the HTML page title tag

**if** (@$pageNavigation) { **echo** "...

If the $pageNavigation variable is set to true, show the navigation bar at the top of the page  
  
File: footer.php

**if** (@$pageElevator) { **echo** "...

If the $pageElevator variable is set to true, insert the HTML page elements to use the page scrolling function

### | URL Hash <-> Search Box Synchronisation: ./js/site.js

**var** elemSearchFilter = document.getElementById("searchFilter");

elemSearchFilter.value = window.location.hash.substr(1);

elemSearchFilter.oninput = **function** () {window.location.hash = **this**.value;}

window.onhashchange = **function** () {

filterSearch.search(hash = window.location.hash.substr(1));

elemSearchFilter.value = hash;

}

This code synchronises the value of the URL hash with the value of the search box.   
*The URL hash is the string after the # sign in a URI (ie VALUE in query.php#VALUE)*

# Work Diary

9/10 - Decide to make a database about my top used software / applications  
10/10 - Create main page (query.php)  
12/10 – Begin populating database content in Excel  
17/10 – Finish populating database content  
18/10 – Export spreadsheet in Excel to MySQL through phpMyAdmin  
23/10 – Create database connection PHP file (dbconnect.php)  
25/10 – Implement category filtering  
2/11 – Create HTML/PHP landing page (index.php)  
3/11 – Create random application function (random.php)  
5/11 – Fix category notifier  
8/11 – Create new entry function (manage.php)  
9/11 – Add images into landing page  
10/11 – Implement entry editing functionality  
13/11 – Code refactoring and directory structure housekeeping  
15/11 – Modify PHP echo outputs to follow HTML code formatting indentations  
16/11 – Implement detailed application view   
18/11 – Modify CSS to look nice (site.css)  
23/11 – Fixed line breaks (\n) not appearing as web page line breaks (</br>)  
28/11 – Fixed manage.php database functions when quotations marks are present in the data of the variables  
29/11 – Implement search functionality into webpage  
1/12 – Implement search ability through the URL hash  
2/12 – Fix search box <-> hash synchronisation  
3/12 – Fix query output to show first record (was accidentally discarding the first entry)  
8/12 – Write project report  
9/12 – Finish project report  
10/12 – Revise project report  
10/12 – Project complete