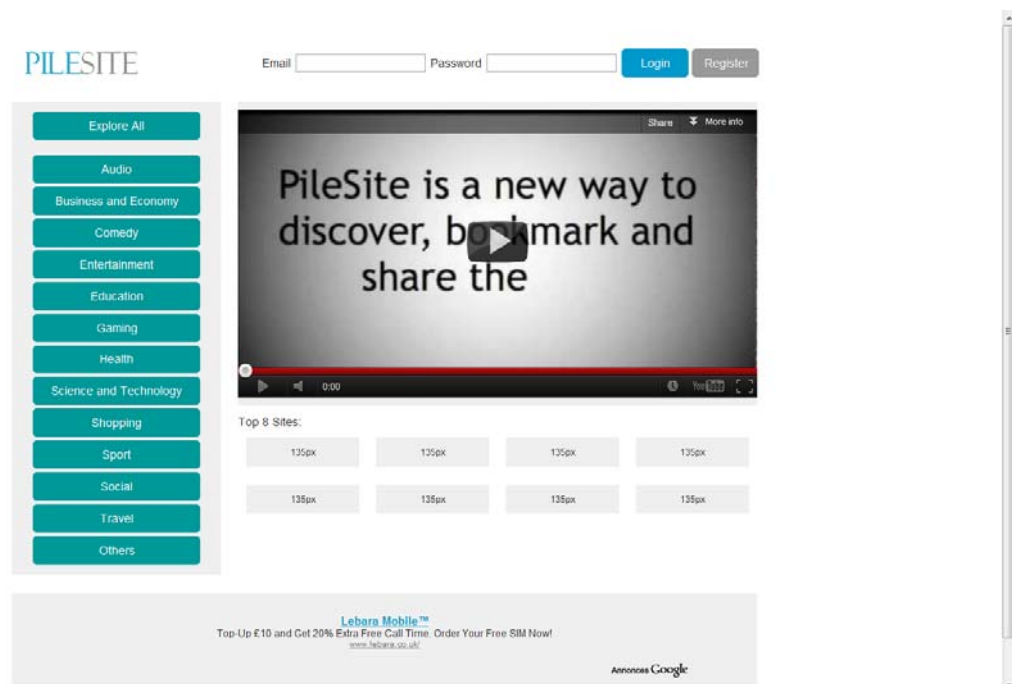# PileSite Project

Team members: Horace Li, Terrence Chiu, Jerrine Soh

Terrence Chiu was responsible for front-end, AJAX and CSS. Jerrine Soh was responsible for video. Horace was responsible for backend MySQL and PHP code.

PHP code fragments taken from PHP 6 and MySQL 5 for Dynamic Web Sites: Visual QuickPro Guide (3rd Edition), by Larry Ullman
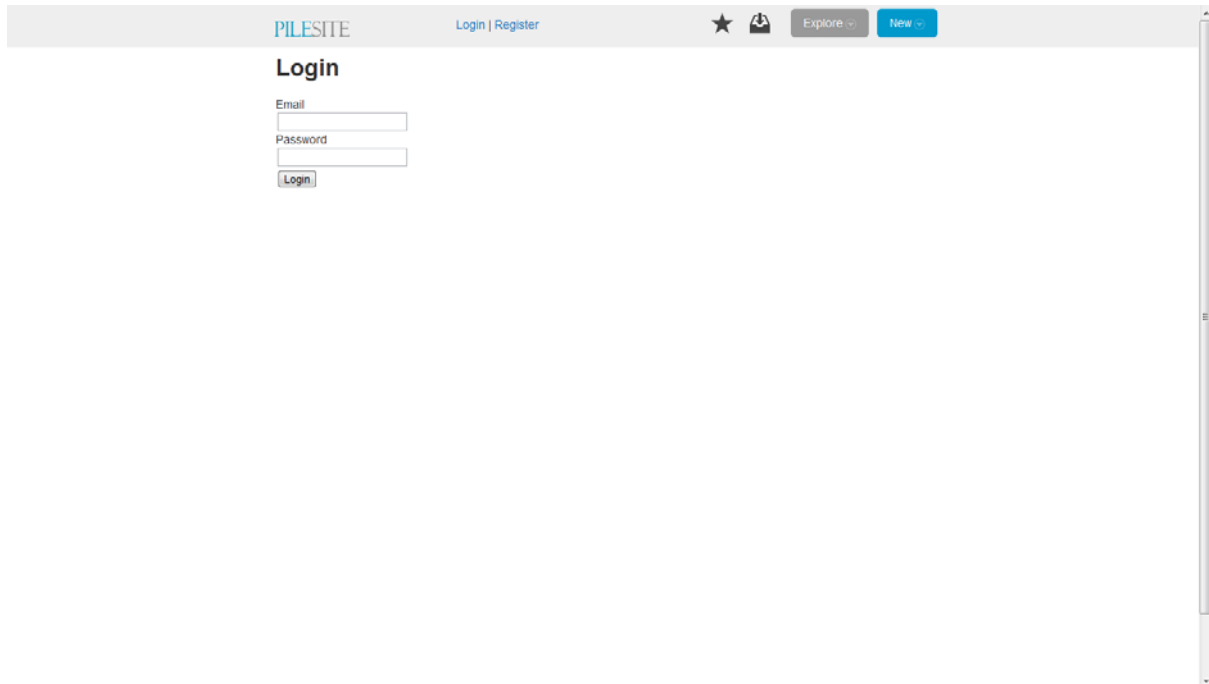
The documentation should either be self-evident by reading the source code or the inline comments, so not all files are reported here.

Index.php

There is very little php, apart for a SQL query that displays the different categories available (left turquoise links). The login button submits the form to login.php.

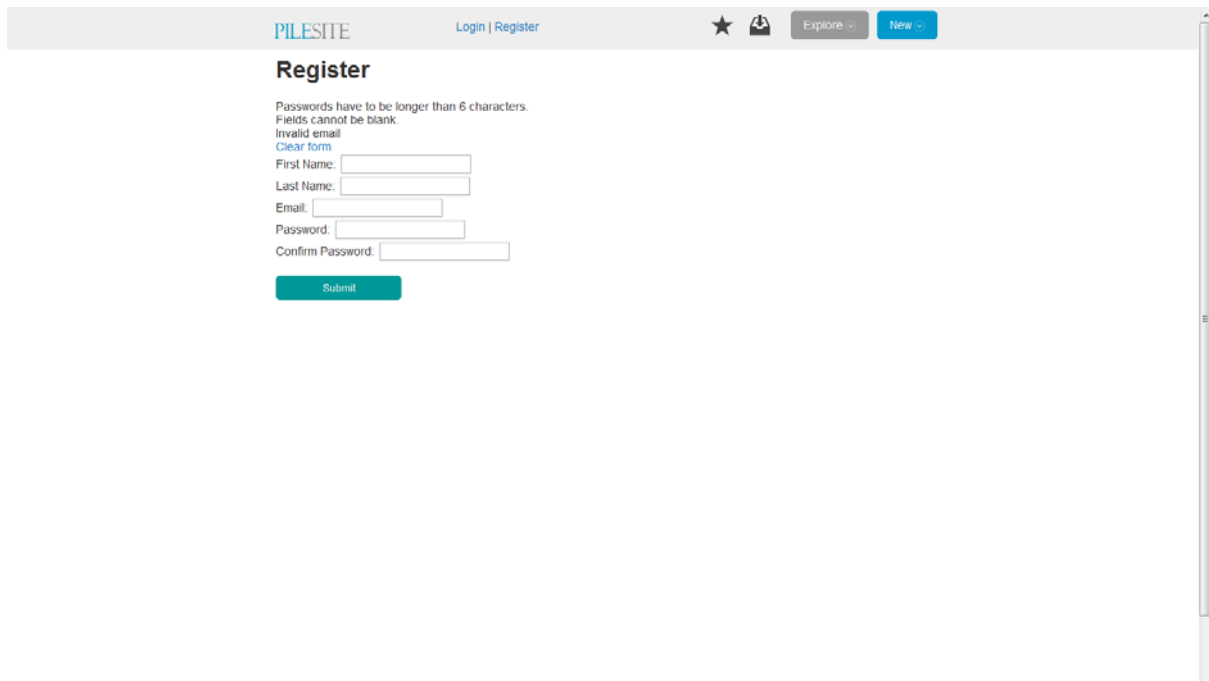## Login.php

In normal usage, this page will only appear if someone types in wrong password or has login errors. It takes POST form inputs and if it queries the database successfully, headers the user to home.php, so normal logins will never see this page.
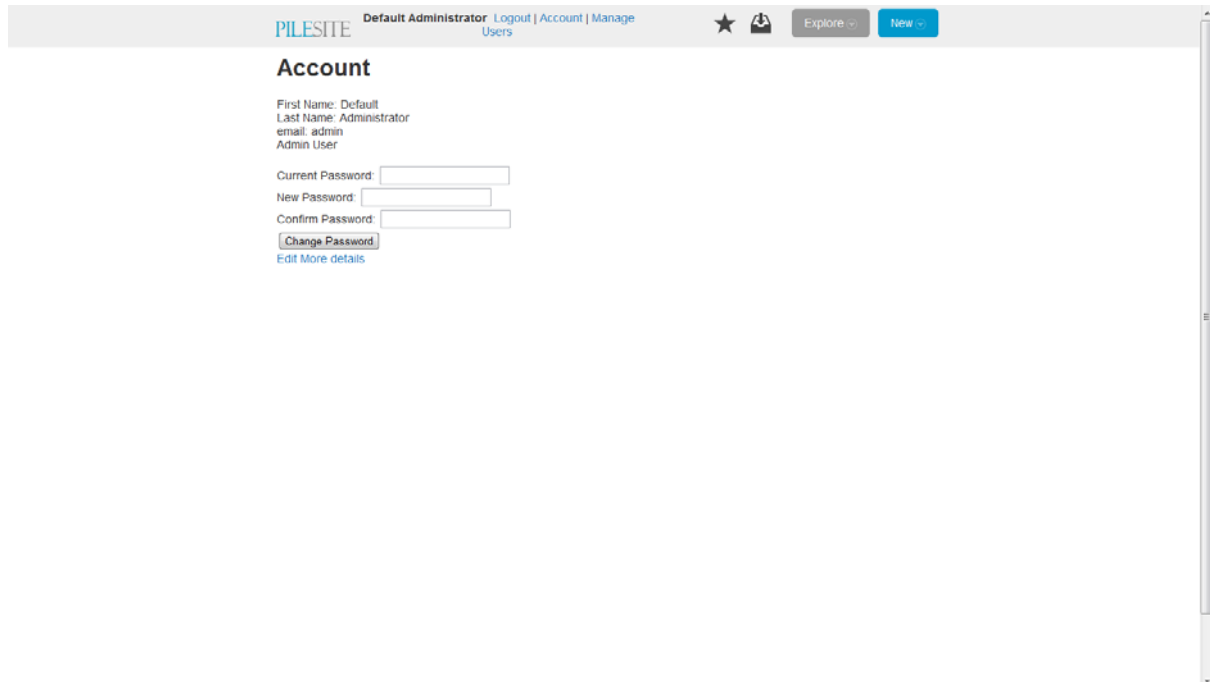
Register.php

The register link on the index page links to this page, which both displays the form and handles it. When handling the form, it all the checks are passed, it inserts the appropriate data into the database and sends an email to the user to activate (see activate.php).

Activate.php

This page has nothing to display. All it does is take the parameters given by the GET array, does a query to check if the data corresponds to a recently registered user, then either modifies the database to 'activate' the user or headers them to login.php which will pick up appended GET values and display the appropriate error messages.

Account.php

This page queries the current user, based on the SESSION user id assigned to the session during login, and displays the users info. It also has a form to change password, which actions back to itself to handle password checking and database update to change the password.

Uploadsite.php

This page shouldn't normally be accessed directly – it is included in the website's global header. The file itself doesn't have any open database connections or css, so it will display an error just accessed alone.

It takes in a website and category values, as well as 2 radio boxes. It first SELECTs the website from the database, and inserts it if it doesn't exists already. Then depending on the values of the 2 radio boxes, it also inserts relevant records into the 'pile' and 'suggestsite' tables. As part of the 'share with others', if the email address of the person you're sharing the site isn't a registered user, it also sends them an email inviting them to register on PileSite.

Warning: mysqli_query() expects parameter 1 to be mysqli, null given in /home/content/55/6261055/html/websites/pilesite/uploadsite.php on line 142

site: [   ]
Category: [▼]

[Submit]

Users.php
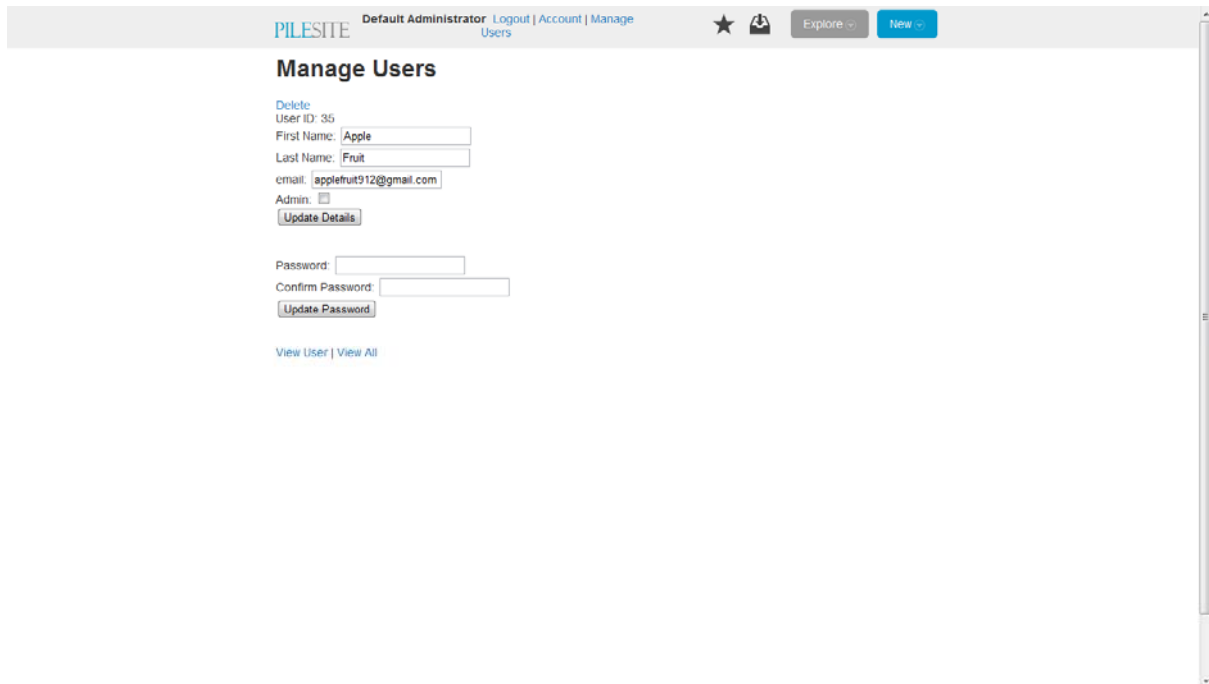

This file headers the user elsewhere if he's not an admin or not logged in. Then depending on the value for the value of $_GET['action'] it includes various other files or forms. If nothing is specified for the action, it includes users_listusers.php. All the files this script includes will have errors if accessed directly.

/includes/users_updatepassword.php

This file checks if the password submitted from the form like

is valid. It checks if the 2 passwords are the same and updates the database.

/includes/users_updatedetails.php

This also takes input from one of the forms in the previous picture. It checks for submitted string length, string already exists in database, and if no errors, updates the database.

/includes/users_create.php

This also only does the form handling. It checks the input for valid strings, queries the database for existing records, then inserts a new record into 'user'.

/includes/users_delete.php

This displays a confirmation message – if you click yes, it goes back to itself, but also appends another GET element called 'confirm', which if detected, will delete the user record from the database instead of displaying the confirmation page.
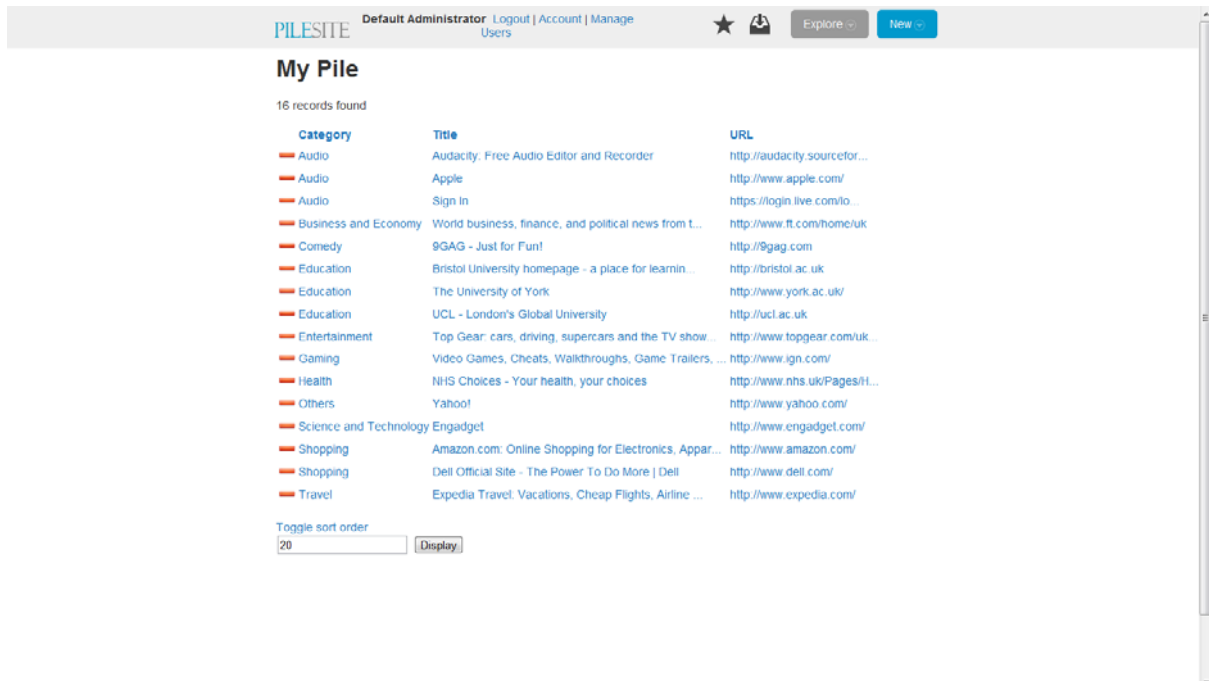
/includes/users_listusers.php

This queries the user table and displays all the users. This based on the example from the book stated at the start of the report.

It first specifies the columns that you can filter by, then goes through the GET values and appends the filter to $queryparameter if the column name is matched. After running the query based on these filters and other sorting options also defined in the GET. It then goes through each column to display the table head, then a while loop going through the database results and again goes through each of the columns to display the required string/value.

If total records are greater than the defined number to display on page, it does some rudimentary pagination. The records to display is set by a form at the bottom of the page.

Home.php

If 'action' in GET is detected and there's an entry id defined, it deletes the relevant record in 'pile' table, otherwise it includes pile_list.php. This displays the sites in my personal pile.
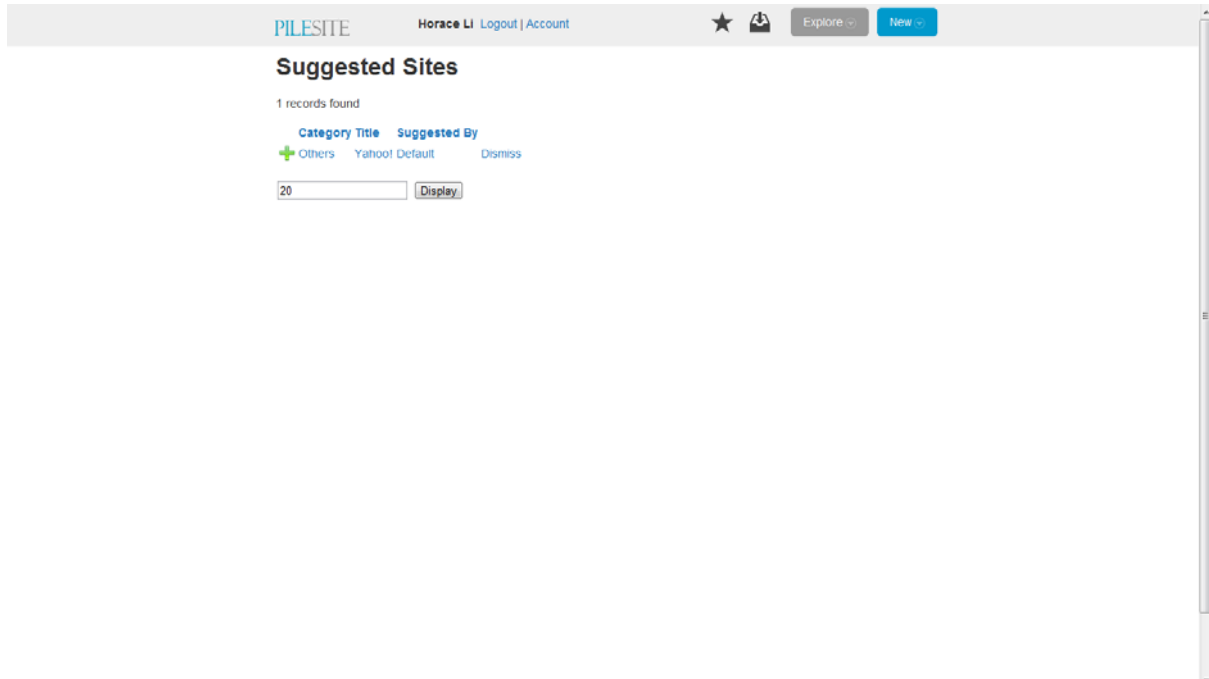

/includes/pile_listpile.php

This is identical to users_listusers.php, except it queries a different table (pile) and the column names and joins are different.
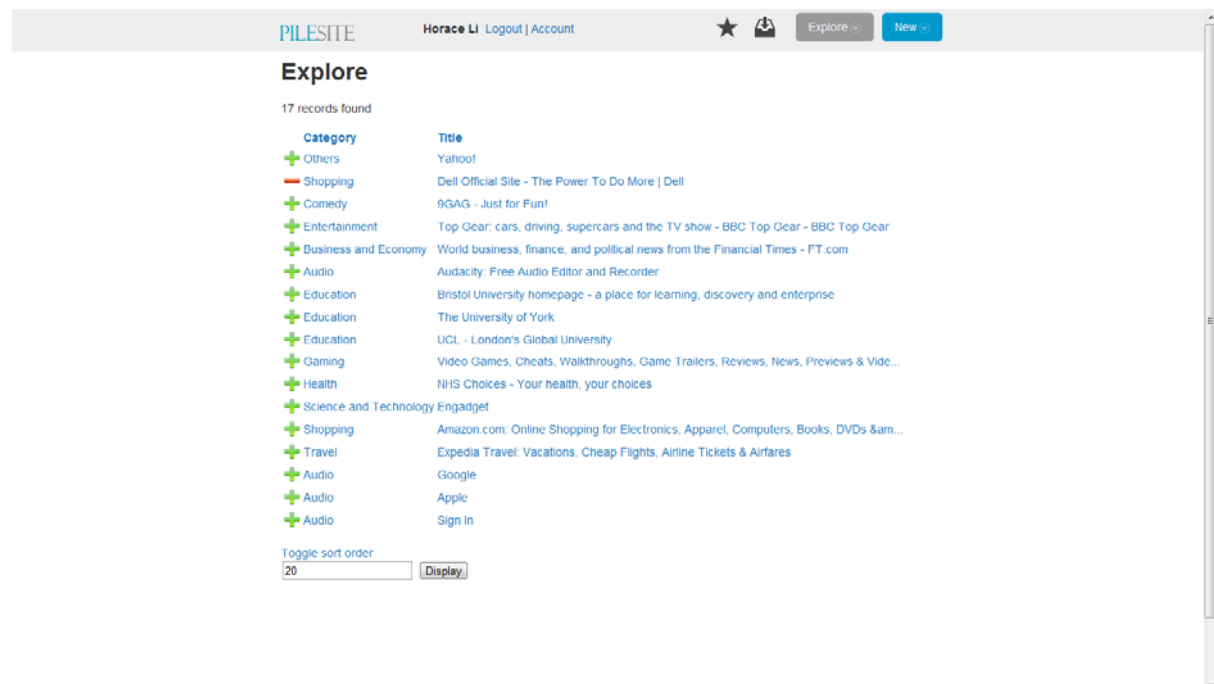
Suggestions.php

This displays sites suggested to me. This is similar to pile.php, but for a different table. The +/- also adds/removes sites from my pile (pile table), but it also adds a 'Dismiss', which removes entries from the 'suggestsite' table. Again, this is handled by checking for $_GET['action']. If no action defined, it includes /includes/suggestions_list.php

/includes/suggestions_list.php

This is similar to pile_listpile.php, but it runs 2 main queries – one on suggestsite table to see what sites have been suggested, another on pile to see if the site's already in my pile (to determine the symbol displayed next to the site +/-).

Sites.php

This can be reached by the explore menu on the header. It lists all the sites in the system. The categories under the explore menu are just links to the same page, but with filters pre applied. It is almost identical to Home.php, but includes a different file to handle the listings. It is also accessible to users who haven't logged in.

/includes/sites_listsites.php

This is similar to to pile_listpile.php, but also runs 2 queries – one to display all the sites that exist and one to check which ones are in my pile.

/global/top.php

This defines functions to regex match email addresses, put a cap on strings too long, and to get the title and URL of websites submitted. It also enables gzip compression, starts a buffer so we can have headers in the middle of files, and makes the database connection.

/global/files_list.php

This has a list of the links to display in the main menu.