Element 010: WEB ARTEFACT Documentation

Laura Marie Cain
Student ID: 1719028

Anglia Ruskin University

BSc (hons) Digital Technology Solutions - Degree Apprenticeship

Fundamentals of Web Development

2018 MOD005616 TRI1 B01CAM

Module Leader: Mark Page

Contents

1. Introduction

Cat Site is a website which displays data related to cats from a database. Users can add their own records and amend records present in the database. There is also a link on the home page which directs the user to a page which displays a video from the external site youtube.com. The website is influenced by the idea of a content management system (CMS), or in this case, a 'cat' management system.

The website is written using a combination of client side HTML, CSS and JavaScript and server side PHP with SQL queries to access database data.

This website uses a MySQLi API to connect to a MySQL database therefore this site is currently not compatible with any databases other than MySQL. The database.php file would need to be amended to use a PDO connection and any reference to MySQLi functions would need to be replaced.

Key Influences:

- Kevin Skoglund for influencing the functions and design of this website. From the Linda.com course 'PHP with MySQL Essential Training: 1 The Basics'.
- Christina Truong for influencing the styling and look of the website. From the Linda.com course 'CSS Essential Training 1'.
- Ray Villalobos for influencing the JavaScript form validation from the Linda.com course 'Validating and Processing Forms with JavaScript' and PHP'.

Look out for **'Key Features'** highlighted in the document.

2. Directories, Functionality and Key Features

The folder structure is initially broken up into public and private folders. The private directory contains files containing potentially sensitive system information. It also contains a shared folder which contains files which remain consistent across the website, e.g. the header and footer. Users do not navigate to these pages directly as they are referenced from other pages. The public directory contains a folder named 'cats' which contains user accessible files (i.e. pages the user navigates to on the website). The public directory also contains folders with a CSS stylesheet, Fonts, Images and JavaScript files.

Private directory

Connect to database: cat_site.config.php

Using the PHP function 'define()', this file defines constants to use for a database connection. The constants include a database server name, database name and database log in credentials.

Database functions: database.php

Accesses the cat_site.config file using 'require_once' to get database constants, then creates a connection to the database. If the connection is unsuccessful, die() is used to display a message to user 'Connection failed'.

MySQLi stores the connection as a variable $db. Database.php also contains user defined functions to check a query set is returned successfully and to disconnect from the database:

confirm_result_set($result_set) The variable $result_set is passed in from database queries (query_functions.php) to make sure SQL query was successful. If no result set is received by the function, then message displays 'Database Query Failed'.

db_disconnect($db) $db variable is passed into this function (from footer.php at the end of every page and from query_functions.php for any failed update or insert query). Checks the $db variable is defined using isset(). If $db is not null then connection is closed using mysqli_close().

User defined functions for general page functionality: functions.php

Key Feature: **url_for($script_path)** Works out if a path is absolute or relative and fixes the path. Uses the constant WWW_ROOT defined in start.php.

- Adds a '/' if not present and is an absolute path (a path starting from system root)
- Removes '/' if present is a relative path. (a path starting from current location)

redirect_to($location) Amends the HTTP header to include 'Location: ' with a passed in URL to make the page instantly redirect to that URL. Headers are sent first in a page before anything is loaded in the page so the redirect happens before the user notices any page content loading. This is used for single page form submission in edit.php and create.php, and to check an ID is passed into edit.php.

is_post_request() Checks if a POST request was made when page loads. This is used for single page form submission in edit.php and create.php.

is_get_request() Checks if a POST request was made when page loads. This is used for single page form submission in edit.php and create.php.

User defined functions for database queries: query_functions.php

find_all_genders() Uses the connection global variable $db (set in database.php). Creates variable $sql to store a string which is a SELECT query which gets data from the 'gender' table in the database. Returns results of query stored as mysqli_result into a variable $result. Checks query was successful by passing $result into confirm_result_set(), which closes the connection and displays an error if this variable is null. This function is also used to display a list from the gender table into a drop down option in the create.php and edit.php forms.

find_all_breeds() Same as find_all_genders() but looks at the breeds table. This function is also used to display a list of breeds from breeds table as a drop down option in the create.php and edit.php forms.

find_all_cats() **Key Feature:** Same as previous two functions but this SQL SELECT query includes two INNER JOINs to the gender table and breed table to show the associated gender and breed name of the cat via a foreign key. This function is used to display cat data in table.php and is also used to pass the associated Cat ID into a GET array for the view.php and edit.php files.

find_cat_by_id($id) $id is passed in from table.php to SELECT data for a row in the cat table and returns the results based on the cat ID. This allows the data to be displayed on the edit.php form and the view.php page.

insert_cat($cat) Using the variable $cat sent in from the associative array from find_cat_by_id($id), this query INSERTs a new cat record into the cat table.

update_cat($cat) Same as above INSERT but performs an UPDATE query to the database.

cat_count() Uses find_all_cats() to get a mysqli_result() variable stored as $cat_set. This is then passed into mysqli_num_rows() to return a value $result which is the number of rows returned in the SQL query, and therefore the number of cats currently in the database. Used to display the number of cats in the database in the header and to see how many options to display in the select drop down in edit.php and create.php

mysqli functions:

mysqli_real_escape_string() is used to escape special characters with '\' to avoid SQL injection in a SQL query.

mysqli_fetch_assoc() gets the result of the SQL query and puts it into an associative array.

mysqli_free_result() is used here to free memory after results are returned from a SQL query.

mysqli_error() displays an error if the UPDATE or INSERT queries were unsuccessful (INSERT AND UPDATE returns a false result from mysqli_result if unsuccessful.

Initialises required files: start.php

Key Feature: Initialises all required functions, global variables, constants and database connections used throughout the site. Turns on output buffering using ob_start(). Defines variable for file paths to PHP constants using dirname() and define(). This is explained in more detail in the file comments. Uses require_once to load and include files; functions.php, database.php and query_functions.php. start.php is accessed via
require_once by table.php, create.php, edit.php, video.php, view.php and index.php. See See Website Design Flow Diagram for visual explanation.

Private -> Shared directory

footer.php

This file is loaded at the bottom of each page for consistency. Shows current year dynamically and then uses db_disconnect($db) to disconnect any active connections to the database at the end of each page.

header.php

HTML elements start to be defined in this file. The <head> tag contains document data including encoding information, a link to the favicon.png shortcut icon, the CSS stylesheet and the page title. The <header> tag displays the page title and images. The header.php file is loaded at the top of each page for consistency

Key Feature: Creates a $page_title variable to display the title of the page starting with 'Cat Site – ' so this dynamically changes depending on the page you are viewing.

intro.php

This file is used to display a banner under the header on index.php and gives a brief description of the page content. Includes the cat_count() to show the number of cats on the database.

This was separated for the flexibility and potential to make the intro section more dynamic by just loading intro.php into each page which then dynamically updates based on the page title.

table.php

This displays the main data table on the index.php with a create cat button and a link to video.php. The find_all_cats() and mysqli_fetch_assoc() functions are used to get array of database data to display in the table. The table also contains an edit and view link for each cat displayed which directs the user to edit.php and view.php respectively.

Public -> Cats directory

create.php

Single page submissions form. Accessed via table.php and accessed via index.php page when user clicks the 'create_cat_button'.

Key Feature: After start.php is accessed, the page checks if a post request. If a post request is not made, then the create cat form is displayed for the user to enter in details to submit data to the database. Once the user submits the create cat form, a POST request is made, data is put into the POST array from the form and then the user is directed back to create.php.

```
<!-- NEW CAT FORM -->
<!-- sends form data to create.php -->
<div>
  <form id="create_cat_form" name="create_cat_form" action="<?php echo url_for('/cats/create.php'); ?>" method="post">
```

Now the page reloads and as a POST request has been made, the insert_cat() query is run to INSERT the data into the database. The page then performs a re-direct using redirect_to() to the view.php page which displays the newly inserted data using mysqli_insert_id(). See Website Design Flow 2 Diagram for visual explanation.

This page links to the javascript file createcat.js which performs validation checks on the form and displays a dialogue box if the user tries to submit the form without entering a valid cat name or age.

Key Feature: Validation checks / actions on forms.

- Display <p> text above the form when user clicks out of the form field if the following criteria are not met:
- Name field is blank
- Age field is not a number (using the JavaScript function isNaN())
- Age field is blank
- Age field is a number greater than 25 (string is converted to a number using the JavaScript function Number() to then check if > 25)

Key Feature: Newly created cat data is provided a default value for the file path field as 'default.jpg'. This displays a generic image which informs the user that the upload image feature is coming soon.

edit.php

Single page submissions form. Very similar functionality to create.php, the page checks if a POST request is made, if not then the form is displayed. Edit.php receives a GET from index.php to display the data in the form for the cat which is being edited. The form is then submitted as POST to and loads itself, which then performs the update_cat($cat) using the data submitted via the form. The page is then redirected to view.php.

```
<!-- EDIT CAT FORM -->
<!-- sends form data to edit.php -->
<div>
  <form id="edit_cat_form"  name="edit_cat_form" action="<?php echo url_for('cats/edit.php?id=' . htmlspecialchars(urlencode($id)));?>" method="post">
```

Key Feature: The image for the relevant cat is displayed at the bottom of the screen using the file_path field in the cats table in the database.

This page links to the javascript file editform.js which performs validation checks on the form and displays a dialogue box if the user tries to submit the form without entering a valid cat name or age.
The validation checks are identical to those explained for the createform.js file, with reference to edit.php not create.php.

video.php

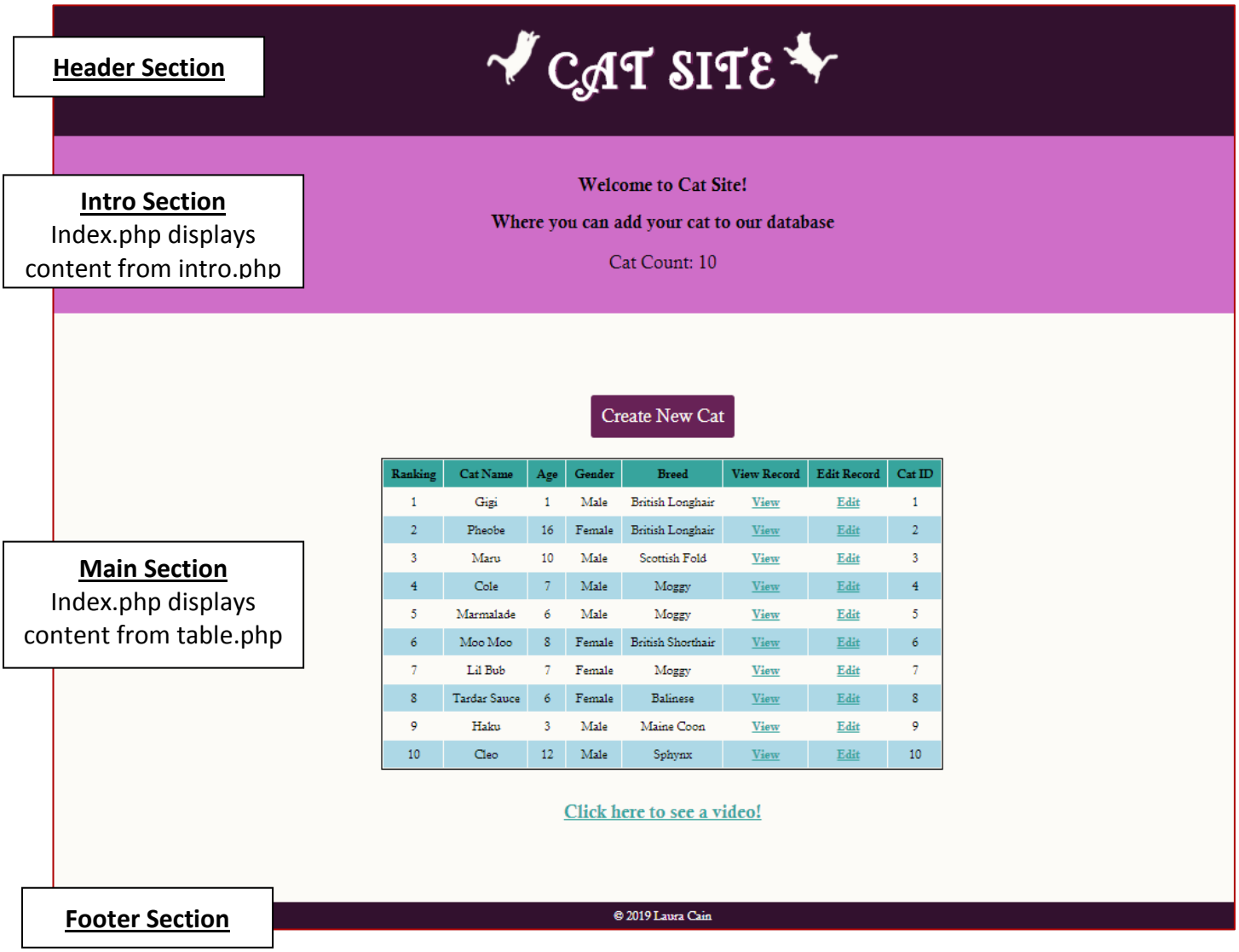
Key Feature: Displays a random video from a variable array of links using JavaScript.

view.php

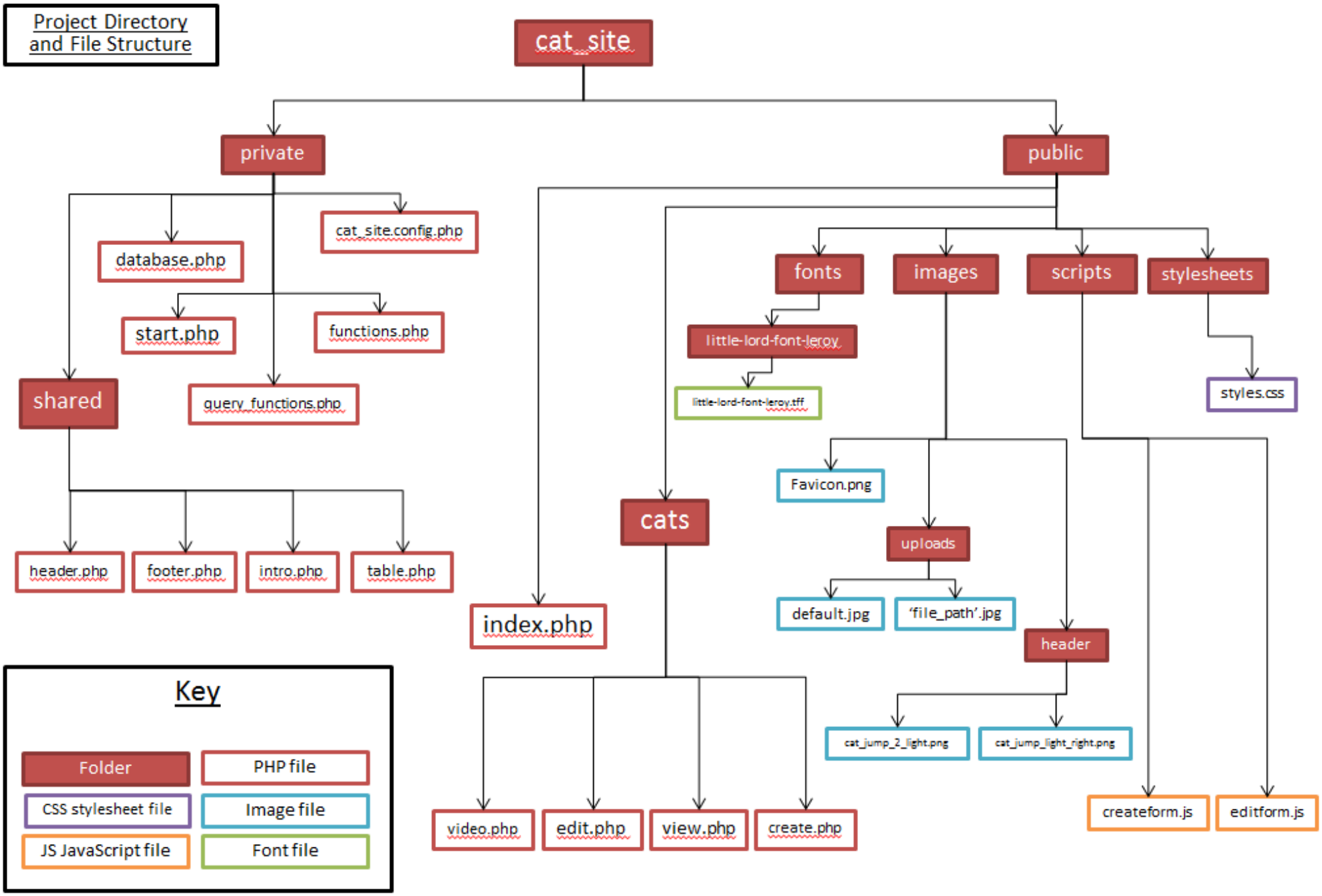
Key Feature: This page displays data relevant to the passed in cat ID. Using find_cat_by_id($id). Like the edit.php page, the image for the relevant cat is displayed at the bottom of the screen.

Index.php

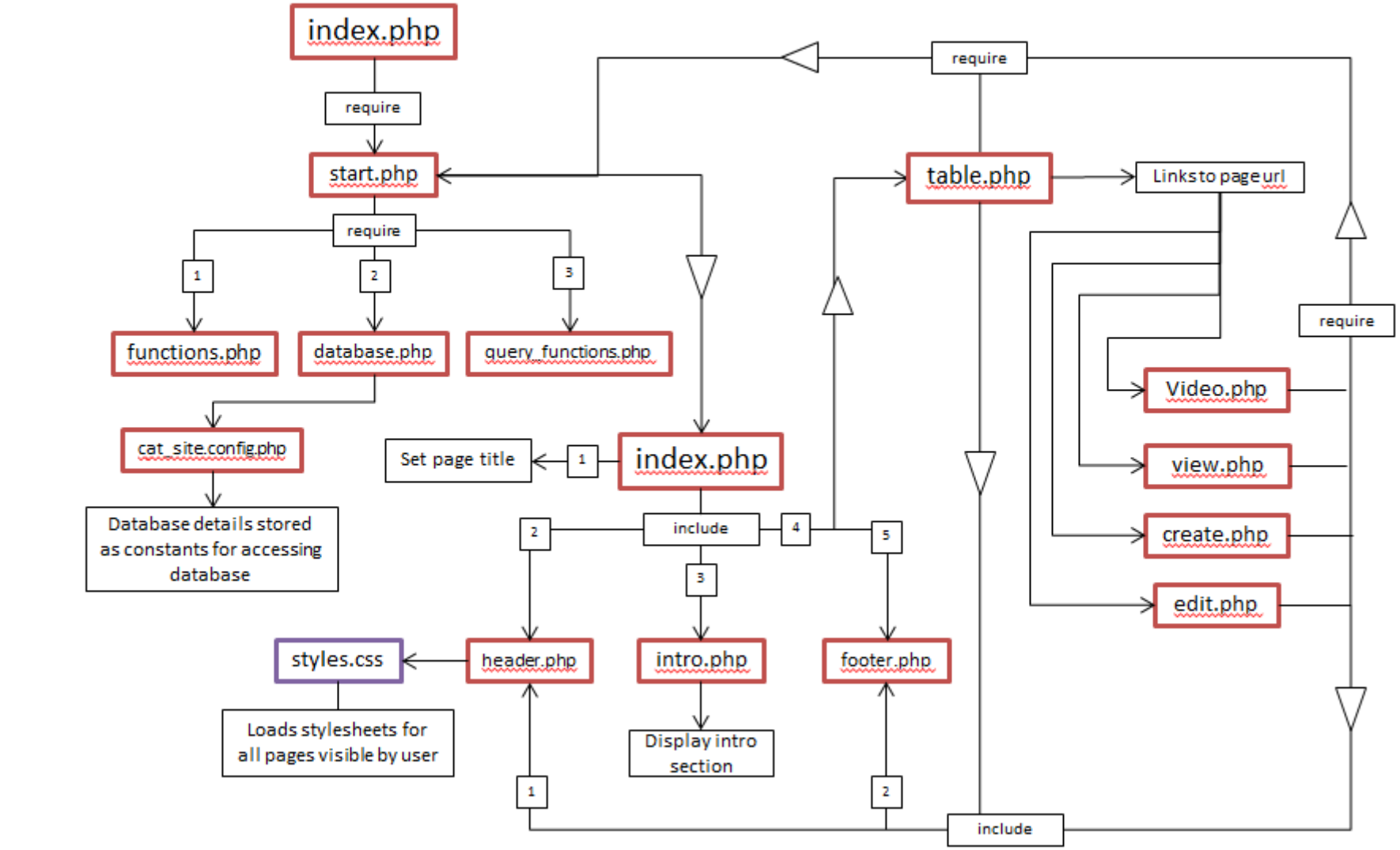
This is the home page. Page title is set to variable $page_title, referred to in the header.php to set the page title depending on which page is loaded.



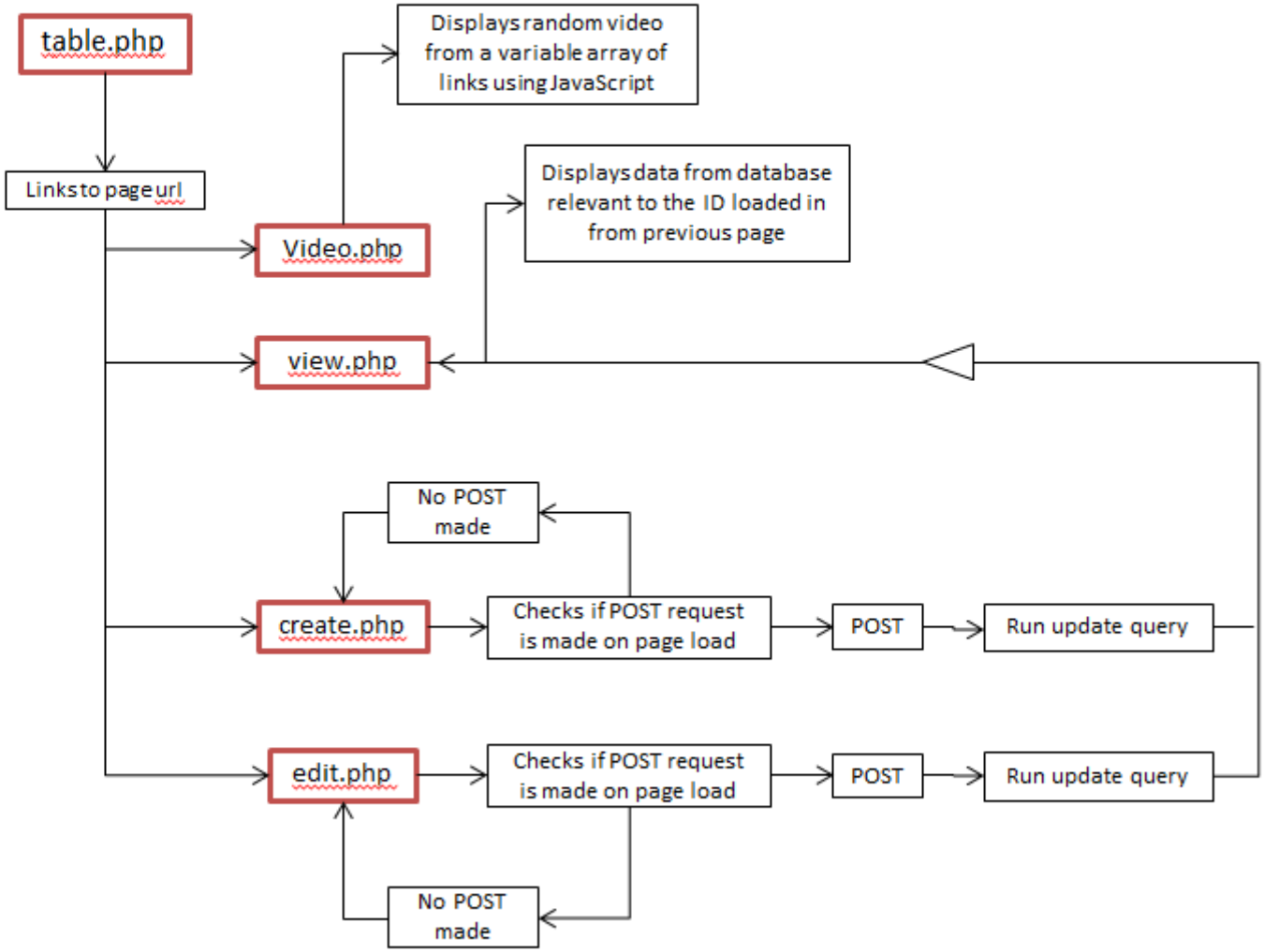
Ranking	Cat Name	Age	Gender	Breed	View Record	Edit Record	Cat ID
1	Gigi	1	Male	British Longhair	View	Edit	1
2	Pheobe	16	Female	British Longhair	View	Edit	2
3	Maru	10	Male	Scottish Fold	View	Edit	3
4	Cole	7	Male	Moggy	View	Edit	4
5	Marmalade	6	Male	Moggy	View	Edit	5
6	Moo Moo	8	Female	British Shorthair	View	Edit	6
7	Lil Bub	7	Female	Moggy	View	Edit	7
8	Tardar Sauce	6	Female	Balinese	View	Edit	8
9	Haku	3	Male	Maine Coon	View	Edit	9
10	Cleo	12	Male	Sphynx	View	Edit	10


Project Directory and File Structure
cat_site
private
public
cat_site.config.php
database.php
start.php
functions.php
shared
query_functions.php
header.php
footer.php
intro.php
table.php
fonts
images
scripts
stylesheets
little-lord-font-leroy
little-lord-font-leroy.tff
styles.css
Favicon.png
cats
uploads
index.php
default.jpg
'file_path'.jpg
header
cat_jump_2_light.png
cat_jump_light_right.png
createform.js
editform.js
video.php
edit.php
view.php
create.php
Key
Folder
PHP file
CSS stylesheet file
Image file
JS JavaScript file
Font file

Website Design Flow



Website Design Flow 2



3. Installation Guide

Install the project files:

1. Access the project directory folder named ‘cat_site’.
2. Files are located in parent project directory ‘cat_site’. Copy the full ‘cat_site’ directory to the localhost directory (normally var/www/html in a linux environment)
3. Test by accessing the path localhost/cat_site/public in a web browser.

Install the database

1. Access the .sql database file named ‘cat_site.sql’.
2a. Option 1. Use built in import tool in phpMyAdmin or Adminer to load the file.
2b. Option 2. Copy the script out of the file and run manually in phpMyAdmin or Adminer.
3. Create a user named ‘student’ with password ‘student’ which has full access to read, create and update all tables in the database cat_site.

Contact

If you experience any issues with your installation please contact Laura Cain:

Laura Cain
07711000579
laura.cain@ground-control.co.uk

5. References

PHP

include / require / include_once / require_once

https://stackoverflow.com/questions/2418473/difference-between-require-include-require-once-and-include-once
https://www.w3schools.com/PHP/php_includes.asp
http://php.net/manual/en/function.require.php
http://php.net/manual/en/function.require-once.php

ob_start

http://php.net/manual/en/function.ob-start.php

define()

http://php.net/manual/en/function.define.php
https://www.w3schools.com/PHP/func_misc_define.asp

dirname()

http://php.net/manual/tr/function.dirname.php
https://www.w3schools.com/php/func_filesystem_dirname.asp

__FILE__

http://php.net/manual/en/language.constants.predefined.php

stropos()

http://php.net/manual/en/function.strpos.php
https://www.w3schools.com/PHP/func_string_strpos.asp

substr()

http://php.net/manual/en/function.substr.php
https://www.w3schools.com/PHP/func_string_substr.asp

PHP $_SERVER

https://www.w3schools.com/PHP/php_superglobals.asp
http://php.net/manual/en/reserved.variables.server.php
https://w3resource.com/php/super-variables/%24_SERVER.php

die()

https://www.w3schools.com/PHP/func_misc_die.asp

isset()

http://php.net/manual/en/function.isset.php
https://www.w3resource.com/php/function-reference/isset.php

MySQLi

MySQLi new connection

http://php.net/manual/en/mysqli.construct.php
https://www.w3schools.com/PHP/php_ref_mysqli.asp
https://www.w3schools.com/php/php_mysql_connect.asp

mysqli_close()

https://www.w3schools.com/Php/func_mysqli_close.asp
http://php.net/manual/en/mysqli.close.php

mysqli_result

http://php.net/manual/en/class.mysqli-result.php

mysqli_fetch_assoc()

http://php.net/manual/en/mysqli-result.fetch-assoc.php
https://www.w3schools.com/PHP/func_mysqli_fetch_assoc.asp

mysqli_free_result()

http://php.net/manual/en/mysqli-result.free.php
https://www.w3schools.com/PHP/func_mysqli_free_result.asp

mysqli_real_escape_string()

http://php.net/manual/en/mysqli.real-escape-string.php
https://www.w3schools.com/Php/func_mysqli_real_escape_string.asp

mysqli_error()

https://www.w3schools.com/php/func_mysqli_error.asp
http://php.net/manual/en/mysqli.error.php

mysqli_num_rows()

https://www.w3schools.com/Php/func_mysqli_num_rows.asp
http://php.net/manual/en/mysqli-result.num-rows.php

Installation

http://php.net/manual/en/mysqli.installation.php

JavaScript

isNaN()

https://www.w3schools.com/jsref/jsref_isNaN.asp
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/isNaN

Number()

https://www.w3schools.com/jsref/jsref_number.asp

Other

https://phpdelusions.net/articles/paths
Kevin Skoglund – PHP with MySQL Essential Training: 1 The Basics Linda.com course
Christina Truong – CSS Essential Training 1 Linda.com course
Ray Villalobos – Validating and Processing Forms with JavaScript and PHP Linda.com course