Photo Gallery Application (FMA): Report

Author: nkatz01 13128128

Date: 06/08/2019

Site address: <http://titan.dcs.bbk.ac.uk/~nkatz01/w1fma/index.php>

**Measures undertaken to protect the integrity of the data stored should failures occur during the uploading or saving of images and image meta-data.**

The following examples are some of the ways in which the integrity of the data stored could have been compromised and the measures which I’ve undertaken to avoid that from happening:

* My code in processForm.php provides for a number of checks during upload, including:
  + whether the file is a legitimate and safe file
  + whether the file is a jpeg type
  + whether the file is already saved in uploads and
  + whether the details of the file are extractable by getimagesize().
* if we fail to insert the details of image into the database we delete it from the upload folder in order to allow for the user to upload it again
* if someone else inserted a row in the database in between our insertion (of a newly uploaded image) and selecting the image for presenting it back to the user, we delete ours from the database and also from the upload folder and we inform the user to try again.
* if our attempt to copy our image to newly resized copies fails, we delete the record associated with that image, already stored in the database (and we also delete the image from the upload folder) to avoid empty img tags
* if we have an image from either thumbs or large folders, deleted manually, we also delete the record corresponding to that image from the database, and also from the upload folder
* if a given record in the database is delete manually, within SQL, we delete it also from the uploads folder
* (if we have an image deleted from the uploads folder, nothing happens)
* to make an image unique, we give every image a time stamp which allows user to upload a different image with the same name

**Brief log of any problems encountered during work and how they were solved (It should also include your report on the application failure points).**

Problem: what if someone queries for an image json details when that image doesn’t exist?

Solution: have the array that populates with the all records returned from the database, record their indices with the record PK and before returning json to the user, point the user provided value at the array to check if it is set.

Problem: how do we get each image on the thumbnail page to link to its larger version and still when we hit at it, and it takes us to the larger version, have the underlying script have access to all the other necessary data and files associated with this image, without having to do includes/requires again or redo all the processing again?

Problem: how do we let the user know what the id, of the image from which they wish to get json of, is?

Solution: build a query string of an array of values and have the link to the larger version of the image link back to index but with an additional query string appended, along with the id of that image, which is now visible to an outside user.

Additional solution to the second problem: on the thumbnail page, have the id of each image appear in the ‘title’ html attribute of an img tag.

Problem: What if a user enters a wrong language preference, eg ‘yo.php’? How do I then output the exception feedback to the browser such that it passes html validation?

Solution: Have a try that checks if the file is valid and have the html template values replaced by empty strings where necessary, within the catch block.

Problem: How to ensure that if a record in the database is deleted, that the corresponding image in the upload folder is deleted too?

Solution: Extract a 1D array $thumbpaths, containing just the ‘filenames’ of all the images stored in the database and another array with all the filenames of the files stored in the upload folder. Sort them in lexicographical order and compare them to make sure each physical file has a corresponding entry in $thumbpaths. If it doesn’t delete it from the upload folder.

**Failure points identified and issues not solved:**

I was unable to figure out a way of how to include all the processing in one try{} block and throw/catch any exception within it, during the entire processing of all the scripts, and at least assign to it customized message and output it to the user, even if together with the other exception messages that php outputs.

I have not reviewed the code enough to ensure that all data is cleaned and escaped where necessary upon database insertion and browser output and that where I have used them, such as htmlentities() or real\_escape\_string(), that they’re always needed.

Even though I have implemented different css settings depending on which format of the image is displayed to the user, I have not been able to figure out how to fix the problem of big gaps on the thumbnail page.

I have not made use enough of generalized functions in my function.php file to aid reuse, bug fixing and encapsulation. I have also not broken my code down in more digestible pieces or files. Instead my code is fairly complex such that scaling it or maintaining maybe quite challenging.

The issue of limited use of semantic (X)HTML and overuse of generic div tags still remains and will need to be worked at in future assignments.