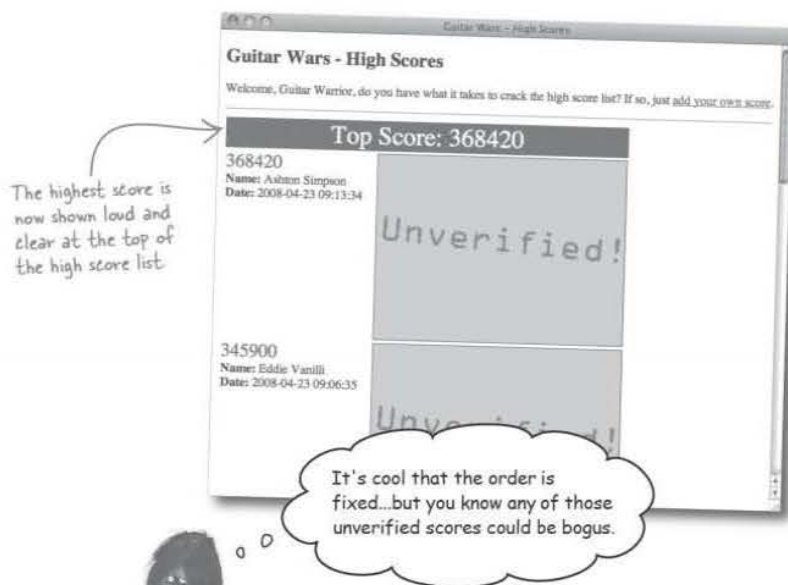




## Test Drive

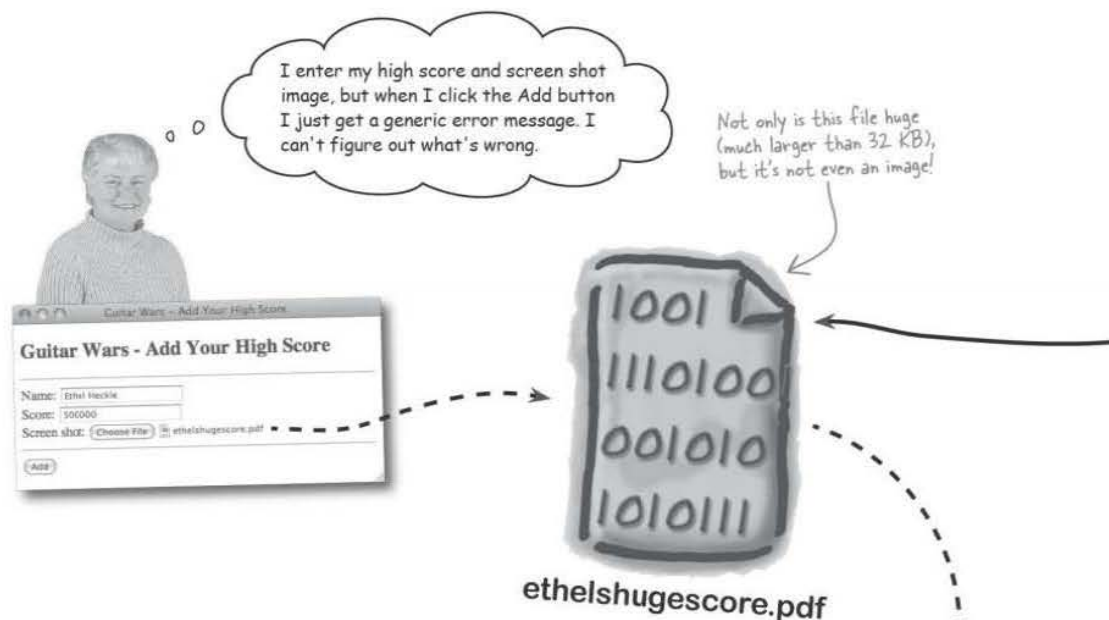
### Order the high scores and showcase the highest score of all.

Modify the `index.php` script to use the new ordered `SELECT` query, and then add in the code that generates the top score header. Upload the new script to your web server and open it in your browser to see the top score prominently displayed.



### It's true, the unverified scores need to be dealt with.

But one thing at a time. It seems another problem has surfaced that is preventing people from uploading their high score screen shots...

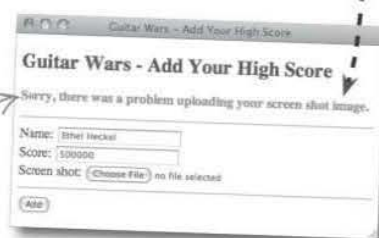


### Not only is the file huge, but it's not an image!

We have a problem in that our form is rejecting some files but not telling users why. It's actually good that the form is rejecting files, in this case because they're too big—remember we capped the file size at 32 KB in the form code. But we need to be clear about telling the user why. Not only that, but we don't want users uploading files that aren't images. Adding validation to the Add Score form will allow us to better control how files are uploaded.

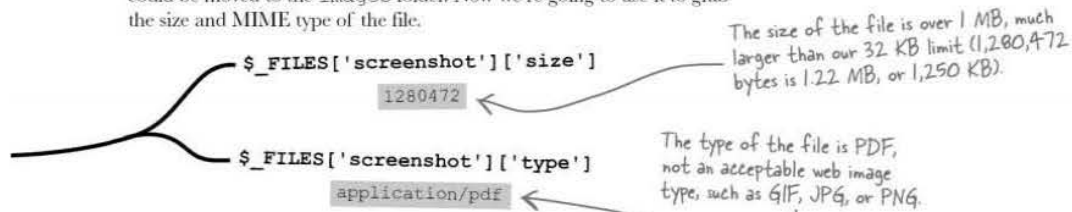
So validation on the image file upload form (`addscore.php`) serves two vital purposes. First, it can beef up the prevention of large file uploads, providing users with notification that a file can't be larger than 32 KB. And secondly, it can stop people from uploading files that aren't images. The file upload form needs validation for both file size and type.

This error message doesn't tell the user much about what went wrong with the high score submission.

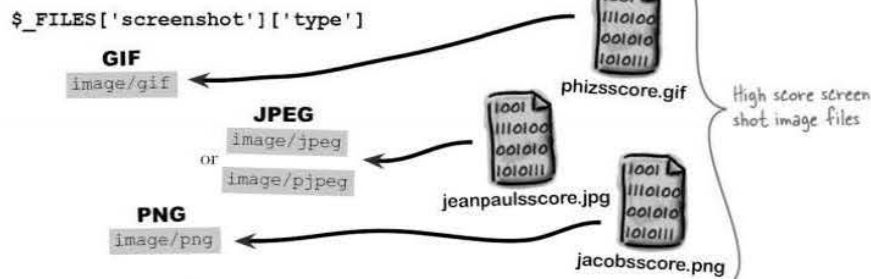


## small Only images allowed

So how exactly do we check the Add Score form and make sure uploaded images adhere to a certain size and type? The answer lies in the built-in `$_FILES` superglobal variable, which if you recall, is where we earlier obtained the temporary storage location of the uploaded file so that it could be moved to the `images` folder. Now we're going to use it to grab the size and MIME type of the file.



We don't just want image files to be smaller than our 32 KB size limit, but we also need them to be a file type that can be displayed as a web image. The following MIME types are commonly used to represent web images:



### Sharpen your pencil

Write an `if` statement that checks to make sure a screen shot file is an image, as well as checking to make sure it is greater than 0 bytes in size and less than the constant `GW_MAXFILESIZE`. Assume the file size and type have already been stored in variables named `$screenshot_size` and `$screenshot_type`.

```
if (
.....
.....
.....
) {
```