Code questions for S.  
  
1. I have a php file that has the connection info to the database, config.php, that is required in any php file that sends or receives data from the database.   
  
Is it safe to have that connection info in that file? The account it uses only has the minimum permissions required to modify the relevant tables. It can’t modify the `users` table. I’m not sure the best way to store that connection info.

2. In header.php, I have code to check if the user is logged in and if not, send them back to the login page. All php files include this header file. Is this the best practice? Does this code actually prevent non-logged in users from accessing files directly?   
  
  
3. In submit\_ticket.php, I have php code attempting to sterilize user inputs from running code. Does this code actually do that? See line, 23, 24, 25 for examples.   
  
  
4. Question: Are you supposed to use php files and just echo out your html code? I saw another way where you end with the php tag, switch to html, then switch back to php when needed? We just did most of it in a <php> tag and echoed out statements.   
  
Answer: You can do it either way. Sometimes it makes more sense to stay mostly in HTML, but jump into PHP for one or two spots, like if there is a long document that just needs a variable echoed here and there. For this tutor-ticket system, though, I think staying in PHP makes the most sense.

5. Question: Is it best practice put the php variables inside the string or do you end the string and start a new one? The former is what I ended up doing.  
echo ‘<p>Hello world ’ . $var . ‘</p>’ VS  
echo ‘<p>Hello world! $var</p>’

Answer: You can do it either way. I almost always end the string and concatenate it with the variable, myself. It feels cleaner to me. But I know some real good programmers who prefer the opposite.

Answers  
1. Is it safe to have that connection info in that file?

The short answer is yes. PHP files are on the server, not the client, so users can't see your code.

Credentials certainly will have to be in some PHP file, somewhere.

What I generally do is define all the "secret" credentials my site will need in one place, like a definitions.php script, that all the other scripts can include.

2. Does this code actually prevent non-logged in users from accessing files directly?

Yes.

3. Does this code actually do that?

Nope. I assume you mean that action attribute is where you thought that would happen?

Your input sanitization has to happen in action\_submit\_ticket.php, the actual script that receives the user's input.

4. You can do it either way. Sometimes it makes more sense to stay mostly in HTML, but jump into PHP for one or two spots, like if there is a long document that just needs a variable echoed here and there. For this tutor-ticket system, though, I think staying in PHP makes the most sense.

5. You can do it either way. I almost always end the string and concatenate it with the variable, myself. It feels cleaner to me. But I know some real good programmers who prefer the opposite.