Uriel Duran

Application Developers

One of the more important parts of building out this website is the ability for users to log in and be authenticated. Since the website revolves around user submitted questions and user answers it is important that users be signed it to be able to access features. Our user authentication will be carried out using our @hawk.iit.edu emails with the help of OTS. We have not yet discussed how this is done but our team currently assumes that it will be something very similar to authenticating by searching through a database of users and matching username/password combination. Once authenticated, there will be different levels of accounts. These levels and their authorization levels are described below:

* Anonymous (Non-Registered Users):
  + Able to read from the database but not write to it. In other words, cannot post questions.
  + Will have access to the post question page but will be prompted to sign in.
* Users (registered users):
  + Able to write to the database (post questions.)
  + Able to delete self-created content.
  + Able to read database (other questions and answers on website.)
* Administrators
  + Able to delete any user content.
  + Can read and write to database.

These features will be implemented using sessions in PHP and the account-specific options (like deleting any post by the admins) will be done through hidden objects via CSS and HTML manipulation with PHP depending on the user.

When a user is authenticated, either via the OTS libraries or our own user DB, PHP will be used to create a session and store the session ID as a cookie via the $\_SESSION[] function. Based on the user we can flag a session as being an admin account or a normal user. PHP scripts will run once the user logs in to check if the user is an admin or a user. If the user is an admin then there will be PHP functions that insert HTML with UI options only available to administrators. For example, each Question and answer page will now display a button in the UI with the ability to execute an SQL query to delete that specific question. While any other user will only have this option if the userID of the active user matches the userID associated with that post. HTML manipulation through PHP is already being used on the current website. Our navigation bar code exists in a separate .php file and every other page has PHP code to load in the navbar code. This way if we need to change navigation bar features we can do It from a single source and not copy and paste it to every page the website has.

Other features being developed is the ability to upload a picture of a bug taken by the user. The image will then be processed via the PHP exif\_read\_data($file) function and some more PHP code will look at the outputted array trying to find the geolocation metadata of the image. If no geolocation is found the image will still be uploaded and displayed in the post. However, if geolocation of the image is found it will ask the user if they wish to add the location. Alternatively, the user can manually enter the location. This location data will be stored in the database and then through Google Map’s API we will plot the location of the bug on a map. The aim of this is to be able to have a sort of “Bug Spotting” feature where users can look up types of bugs and they can see on the map the different places they have been found.

Similarly, another important feature is the ability to search. There will be a search bar in the navigation bar where a user can search for keywords. A PHP function will then submit a SQL query to the database searching through the questions or comments and will display back a list of where the keyword was found. We are still finding out the best way to do this, looking to see if there are any existing search algorithms we can use because as we currently have it the display will simply be a list of places where the string being searched for may appear even if the question has not little to no relevance.