Blogalert Implementation Summary

COSC 360 Final Project – Anamica Sunderaswara & Laura Letef

The website “Blogalert” is a stylized and completely functional blogging website that allows registered users to create blog posts, leave comments and access profile information. Unregistered users are able to view blog posts and comments in real time, but cannot blog anything themselves or leave comments until they have logged into their account. Blogalert also supports multiple user usage. The final user classification included in this blog website is the ‘admin’ status (usernames that contain the word “admin”). These users are able to access the “Admin Controls” on the profile page *(Appendix A).* From this page, administrative users can search all users in the Blogalert database. After retrieving a list of users from their search, administrative users have the option to delete a user, which will additionally delete all of the user’s posts and comments.

The user functionality features implemented through the Blogalert website include the following:

- Users are able to browse blog posts and comments without registering for an account
- Users can search for posts by keyword or username without registering for an account
- Users can create an account by providing their full name, email, username, profile picture, and a password (which they are asked to confirm)
- If users create an account with a username that already exists, an error message is shown
- Users can login by providing a username and password
- If the account does not exist, users will not get past the login page
- Only once logged in, users can create posts and leave comments on their own blogs or other user posts.
- Users are able to view their profile information, as well as, all blog posts and comments left on their posts
- Users are able to edit their profile by deleting posts that they have made
- As a website administrator, admin users are able to search for user by their username
- Admin users are able to delete a user, which will delete their posts and comments too
- All users are able to log out at any time, bringing them back to the home page.

The general website functionality features implemented through the Blogalert website include the following:

- A hand-styled layout with contextual menus indicating that a user has switch to a new page and a 2-column layout that heavily utilizes a responsive design
- Form validation using JavaScript and database storage using PHP and MySQL
- Server-side scripting with PHP
- Password hashing
- Adaptable design for various screen sizes
- Users are remembered once they login as they navigate from page to page
- User profile information, including profile image is stored securely in database
- Users are easily able to determine which page they are on using thorough page labelling
- Users can easily navigate to any page by using the navigation bar
- Users are often prompted with a “back to ____” option to handle errors with navigation
- Users can search for specific topics, usernames, or keywords

- Blogalert contains various styling flourishes, including personalized pages for increased UX
- Users can track their comments through the profile page
- All pages are styled following specific Blogalert branding

Essentially, users are prompted to select whether or not they would like to sign in when they first open the site. From their selection, the goal of the website is to produce blog posts, whether these be simple one-liners or paragraphs. Users are able to connect with other users on the Blogalert website by commenting on each other's posts. A user can search for specific blog posts either by keyword or username. This website also allows for customization as users are able to view their own profiles, past posts and comments, and delete them from the database accordingly. This is all achieved through the following files:

home.php: This is where Blogalert begins. Here, users are prompted to select one of the following options: login, signup, or "I think I'm good" which is for the users who do not want to login or register at this time.

login.php: If users select login from *home.php* they are brought to this page. Here users are prompted to login using a username and password. For error handling, users can select an option to sign up instead or to return to *home.php*. If the user does not exist, the user will be stuck on this page until the correct information is entered or another navigation option is selected.

loginprocessing.php*:* This page connects the login page to the database and ensures that the users login credentials are correct. If the information that the user enters does not match with what is in the database, an error message is displayed.

signup.php*:* If users select signup from *home.php* they are brought to this page. Here users are prompted to create an account using their full name, email, username, profile picture, and password. For error handling, users can select an option to login instead or to return to *home.php*. If the username that is entered already exists, users are brought to the *signuperror.php* page. Once a valid account is created, users are brought to the *login.php* page to log in. Users cannot click the signup unless all fields are filled in, except for the profile image.

signuperror.php*:* This page displays a personalized message for those who enter a username that already exists. A message will appear indicating users to navigate back to the home page.

frontpage.php: This is what is referred to as the "feed" for Blogalert. Here, a user can find all of the recent posts that have been made by all users in the Blogalert database. A user, registered or not, can search through the recent posts via keyword or username. If the user *is not* logged in, they only have the capability to view the recent posts and comments. If the user *is* logged in, the user has the ability to create a new blog post. This text box will appear at the top of the page. Alongside these features, users can also access the navigation bar on the top right area of their screen, allowing users to navigate to their profile page or logout. If the user is not logged in, the navigation bar will prompt users to log in or go back to the homepage. If a user selects, "view comments" on one of the recent blog posts, they will be taken to the *comments.php* page.

comments.php: The *comments.php* page is a way to display all of the comments made to a single blog post. Here, registered users can read the blog post they selected, who wrote the post, read the comments already added to the post, or add their own comments accordingly. As a nonregistered user, the blog post, author, and comments can be viewed, but unregistered users cannot leave any comments themselves. The navigation bar will indicate navigation back to the feed page.

profilepage.php: This page is only accessible to those who are registered. If an unregistered user somehow finds themselves upon this page, they will be immediately taken to the login screen. The profile page displays a personalized message at the top of the screen to welcome the user, followed by their inputted profile picture. Immediately below the profile picture, a registered user can view all the blog posts they have written as well as two options below each post: view the comments left on the post or delete the post. Viewing the comments will take the user to the *comments.php* page and deleting the post will immediately update the page to show the new list of user-created posts. There is no need to refresh to see this change. The navigation bar displays log out and feed, allowing users to navigate back to the *frontpage.php* page. If the user is an admin, the "admin controls" button can be seen at the bottom of their blog posts (*Appendix A)*

finduser.php: This page is accessible to administrative users *only*. Here, an admin can search for a user by username in the prompted search bar. After clicking the "find user" button, admin users will be brought to the *finduser.php* page. The navigation bar on this page displays three options: log out, feed, and the ability to navigate back to their profile page.

finduser1.php: The administrative "control panel" is referred to as this page. This page displays the entered user that the admin searched for and all of their information, including their first name, last name, email address, user ID, and profile picture. The button at the bottom of the page prompts the administrative user to delete the user that was returned. The navigation bar on this page displays three options: log out, feed, and the ability to navigate back to their profile page. If a username is entered incorrectly in the *finduser.php* page or the user entered does not exist, this page will return the error message "user not found" and a navigation option for admin users to try the search again, which will bring them back to the *finduser.php* page.

deletepost.php: This page is the back-end code that connects to the database and deletes the blog posts created by the user that would typically appear in the profile page view.

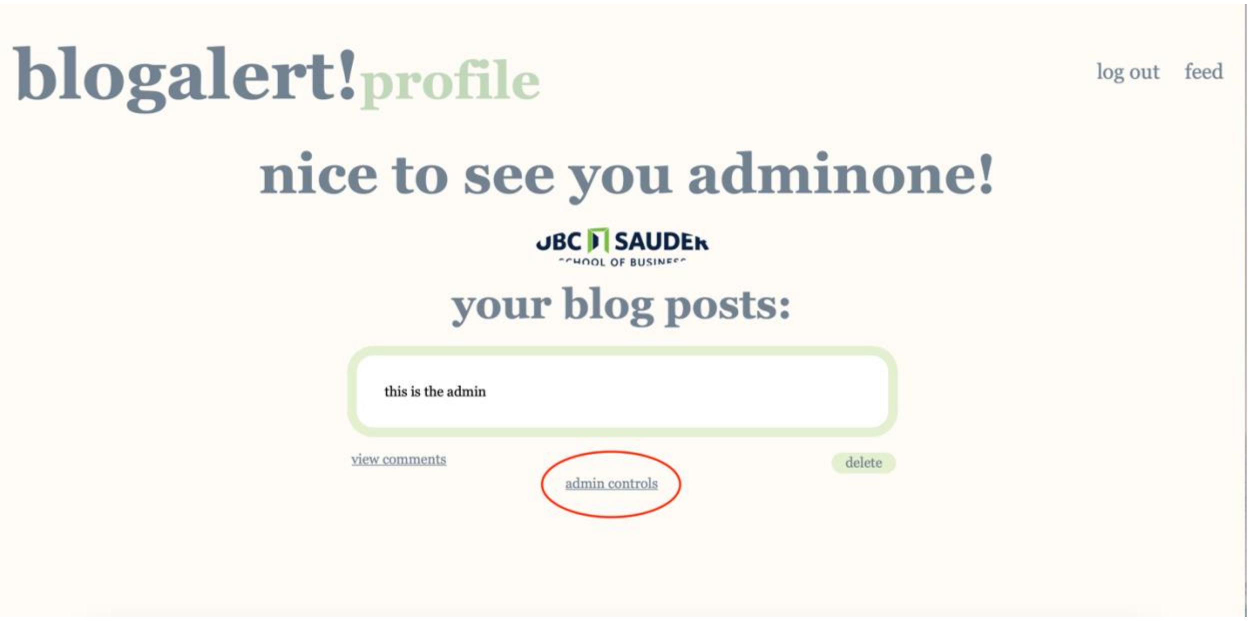
deleteuser.php: This page is the back-end code that connects to the database and deletes the user and all of their posts, images, and comments from the database.

successfuldeletion.php: This page verifies that the user selected by the admin has in fact been deleted and removed from the database.

SignupTest.php: This file tests the sign up page to ensure every field is working correctly and the database is accessed properly & securely.

One of the main limitations of Blogalert is that a user cannot change or modify their profile. In future developments, this is a high priority to fix in order to increase UX/UI design.

Appendix A: Admin Control View



nice to see you adminone!

your blog posts:

this is the admin