

The University of British Columbia Okanagan

COSC 360 - Web Programming



Milestone 3

Final Documentation

Andy McDonald -10817245
Noah Ward - 86636347
Livia Zalilla -19091479

Due Date: December 7th, 2021

Executive Summary

This report documents the work completed to date towards our web-based discussion forum: The LAN Connection. Featured are the most essential features of the site and descriptions of our PHP displays pages, PHP functional files, and JavaScript files. The high-level description is added to show how the use of a relational database and state variables are passed between pages. The known limitations of the site are acknowledged due to the time constraint of the project. And rounding out the report is a complete user guide for an unregistered user, a registered user, and an administrator.

Table of Contents

Major Features	3
File Descriptions	3
PHP Display Pages	3
PHP Functional Files	4
JavaScript Files	6
High Level Description	6
Known Limitations	7
User Guide (Walkthrough)	8
Section 1: Unregistered User	8
Section 2: Logged-In User	9
Section 3: Logged-In as Admin	11

Major Features

As a discussion forum website, the most essential features of the site consist of the accounts and posts. Users can navigate the site to view outdoor adventure conversations in a multitude of categories or search for specific posts using keywords. They can create an account with their personal information and profile picture if desired, and make edits to their password or profile picture whenever from their profile page. Creating an account allows them to create their own posts and comment on others. The site also features a bookmarking system for users to save their favourite posts, and a like system for users to see what posts are most popular. Administrators of the site have the ability to search for users through the administrator portal and can view profiles, disable and enable user accounts, delete posts, and delete entire user accounts.

File Descriptions

PHP Display Pages

admin-portal.php: This page displays the admin portal and is only accessible by admins. This allows an admin to search for users and perform actions on their accounts.

bookmarks.php: This page is for signed-in users only. This page is displayed when a signed in user clicks on the bookmarks link. There are two possible bookmark links: one on the home page and one on the profile page of a signed in user.

category_page.php: This page displays the category page of the category the user has clicked on. It dynamically updates to show one of the 6 possible categories. It does this through a get request contained in the link the user clicked on to navigate to the page.

home.php: This is the home page. This is navigable by a non-signed user, it displays the categories, the most recent posts and a create post box. Although only a registered user can create a post.

post.php: This page is navigable without being signed in. This page is displayed when a user clicks on the comment icon of a post. That post is isolated by its post id and displayed on its own, along with all its comments. A registered user can add their own comment here.

profile.php: This page is navigable by a non-registered user. This page may be displayed when a user clicks on their own profile link in the nav bar or by clicking on a user's username on a post. It displays the user's information and all their posts and on a user's own profile, they may change their profile picture, change their password, create a new post, or access their bookmarked posts.

searchResults.php: This page is navigable for a non signed in user. This page is displayed when a user enters keyword(s) into the search bar located in the navbar at the top of any page.

PHP Functional Files

bookMarkSystem.php: This php file handles two user actions: adding a new bookmark and deleting an existing one. In the first case it adds the signed in user's bookmark information to the database, and in the second case it deletes the tuple from the database.

changePassword.php: This php file handles the changing of passwords: it checks that the old password matches that in the database, validates the new password, and then updates the system with the new password.

changePfp.php: This php file handles changing of users profile pictures: it checks if the given password matches that in the database and then uploads the new profile picture to the database.

checkAdmin.php: This php file handles the checking of admin: it checks if the signed in user is an administrator.

connectDB.php: This php file stores a function that is called whenever a php file needs to connect to the database.

createComment.php: This php file adds a user's comment information (comment date, profile id, comment text, username) to the database. It handles any exceptions along the way: if a user is not signed in, or if a field is left blank in the post, etc.

createPost.php: This php file creates a post and adds the post information to the database. It also handles exceptions where a user is not signed in, or if a post submission or post category is left blank in the post, etc. When creating a post, if an apostrophe is included in the post body, it is converted and inserted into the database as a tilde (~). This is to avoid errors on insert into the database and to avoid SQL injection (could be hashed though to improve security). It is converted back to an apostrophe when the post or comment is displayed

deletePost.php: This php file deletes a user's post from the database using a simple query.

deleteUser.php: This php file deletes a user and all of their related data including their likes, bookmarks, posts and comments from the database.

disableUser.php: This php file disables a user account from the database: it checks if the user account is enabled or disabled and toggles the state in the database.

displayPost.php: This php file outputs the html to display a post: it displays the number of likes, dislikes, and comments. It checks if the signed-in user has liked, disliked, or bookmarked the post. It also displays the delete post/user buttons if the signed in user is an administrator.

handleImg.php: This php file contains two functions: one for uploading an image to the database and one for accessing an image from the database. Its built to handle both images for profiles and for posts, although the latter was never fully implemented.

likeSystem.php: This php file receives a post request from likeSystem.js containing the post id and action. This file will update the database depending on the action, which there are four of: like (insert a row into the database), unlike (delete that row from the database), dislike (insert), undislike (delete). This is handled by a switch statement. This file also determines which user is signed in, and does not execute if there is no signed in user. It also ensures a user can only like a post once, and cannot have a like and a dislike at the same time in the database.

login.php: This php file logs in a user and handles a range of scenarios including invalid request method, invalid username, email, or password, etc. It then updates a state variable for the signed in user upon successful login.

logout.php: This php file is called when a user chooses to log out.

navBar.php: This php file displays a navigation bar with a link to the homepage, category links when on category pages, login/sign up links for unregistered users, profile/logout links for registered users, and an admin portal link if the user is an administrator. This is included in many of the PHP Display pages.

newUser.php: This php file creates a new user and adds them to the database. In the process, this file handles all possible events: invalid request method, missing fields, existing users with given username or email, etc.

retrieveLikes.php: This php file checks if a user has already liked, disliked, or bookmarked a post. It stores functions that return the number of likes, dislikes, and comments on a post. This is called for every displayed post in order to display this information. It also determines if the signed in user has already liked/disliked a post, and if so, it updates the like/dislike button styling. This allows the user to know if they have already liked/disliked/bookmarked the post.

validateText.php: This php file cleans user inputted text for consistency purposes. It strips spaces, slashes and converts special characters to HTML.

JavaScript Files

bookmarkSystem.js: The purpose of this JS file is to asynchronously update the page to bookmark or un-bookmark posts. It will update the bookmark icon to become filled in (solid colour) upon clicking so that the user knows their bookmark has been successful. If the icon is already filled in, clicking the icon will change it to be hollow so the user knows their un-bookmark has been successful. Based upon these two actions, this JS file will post the action and the post id to a php file called bookmarkSystem.php.

dropdown.js: This file displays a three-dot icon on user posts if the signed in user is an administrator. It gives the admin the option to either delete a user's post or their whole account. Either of these options if clicked triggers a confirmation message where the admin must confirm this is the action they wish to perform.

likeSystem.js: This file provides like and dislike functionality (like and dislike posts). This file contains two functions: clickedLike(obj) and clickedDislike(obj). Each function is called when a user clicks a like or dislike button, respectively. Each function will update the like button to be filled in (solid colour) upon clicking so that the user knows their like has been successfully recorded. If the like button is already filled in and the user clicks it, the action will be to 'unlike' the post. Possible user actions include: 'like', 'dislike', 'unlike', and 'undislike'. Based upon the user action and button click, the JS file will post the action and post id to a php function called likeSystem.php.

sortPosts.js: This asynchronously changes the style of the post sort buttons upon click. These sort buttons were commented out as time to implement this entire feature ran out. It is prepared to send information to a php file that would return the displayed posts in a new order depending on which button was clicked.

High Level Description

The site works through the use of a relational database and state variables passed between pages. When a user logs into the site, a state variable is updated to hold who is logged in. This is how some functions are hidden from unregistered users, as well as allowing the user to perform multiple actions without needing to sign in again and again. All pages access the database to populate themselves, whether to display and add posts, or to sign up and login as a registered user. Most user interactions involve a form that sends information to one of the many PHP files that validates it before using it to affect the database. Once an action is done on the database, the user is redirected back to a display page of the site. Many of the functional PHP files consist of PHP functions or code that is brought into other files in order to limit duplicate code across pages. Server side scripting is used at times to allow database updates without a page reload being required, such as with bookmarks and likes.

Known Limitations

There exist a handful of limitations and many if not all are simply due to time constraints. Although planned, the ability to display images as part of posts did not work and as such the code for this was either commented out or fully removed in places despite the database being prepared to hold images related to posts. The ability to sort posts by likes, dislikes or date was also planned but never fully implemented past styled HTML buttons prepared to perform asynchronous updates using JavaScript. The HTML for the buttons were commented out. Administrators are able to enable and disable users from the administrator portal. When a user is disabled they are unable to login. This is limited by the fact that if the user is already logged in they may stay logged in and contribute to the site and only once logged out are they unable to log back in again to make further changes. Additionally, a limitation to the user is that they are unable to create their own categories to post to and as such are limited to the categories that are present already. The security is in places potentially lacking and provided more time would be seriously investigated to fill in any cracks.

User Guide (Walkthrough)

Section 1: Unregistered User

1. Home Page

The screenshot shows the LANconnection homepage. At the top, there are search bars and a login/signup button. Below that, there's a section for 'Categories' with thumbnails for Mountain Biking, Hiking, Climbing, Snowboarding, Golf, and Hockey. To the right, there's a 'Popular Posts' section with four posts by users 'andymot', 'nothward', 'tendybrendy', and 'jakeywill'. Each post includes a timestamp, a 'Post to' category, and a brief description. A 'Personal' sidebar on the right allows users to 'Create a post'.

From the home page, you have access to the different categories, the most recent posts, the post search bar, and the login/signup page. This page can be returned to at any time by clicking the home icon in the top left corner of the screen.

2. A Category Page

The screenshot shows the 'Snowboarding' category page. It features a header with a search bar and a login/signup button. Below the header, there are thumbnails for various snowboarding posts. A central banner says 'Snowboarding'. A 'Post to' field is present at the bottom of the page.

By clicking on a category, you can view all the posts posted to that category. On any post, by clicking on the third icon (a little speech bubble) you can go to the post's page and view any comments on said post. The number of comments appears beside the icon on the post itself.

3. A Post Page

The screenshot shows a post by user 'nothward' titled 'Snowboarding'. The post has three comments from 'andymot', 'tendybrendy', and 'coleweed'. Each comment includes a timestamp, a 'Post to' category, and a brief message. A 'Comment' form is located at the bottom of the post.

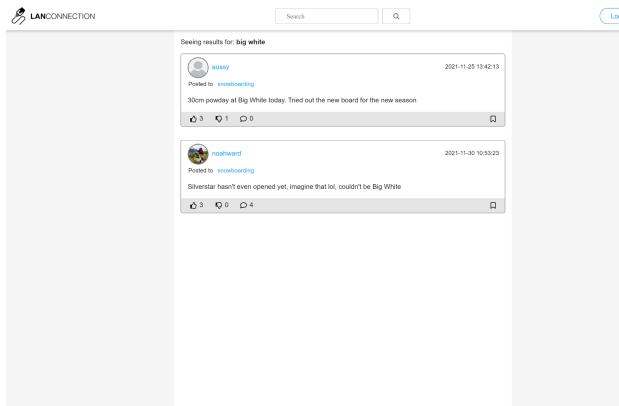
On all posts, the profile of the poster and all the commenters can be accessed by clicking on their names.

4. A Profile Page

The screenshot shows the profile page for user 'nothward'. It displays three posts by 'nothward' with timestamps and descriptions. The user's profile picture is shown on the right, with the name 'Noah Ward' and the handle 'nothward' below it.

When visiting a user's profile page, you can view the posts they've posted, as well as their full name and username.

5. Search For Posts

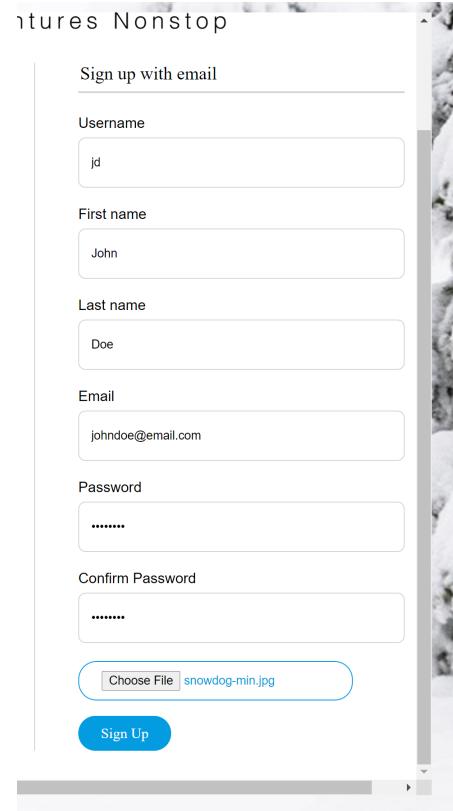
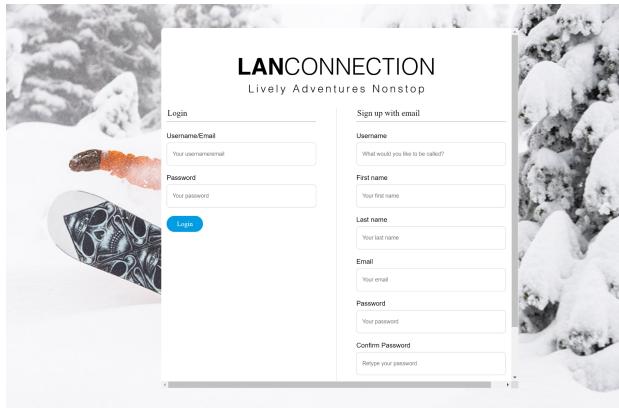


From any page you can access the search bar at the top, and search for posts by their content.

Section 2: Logged-In User

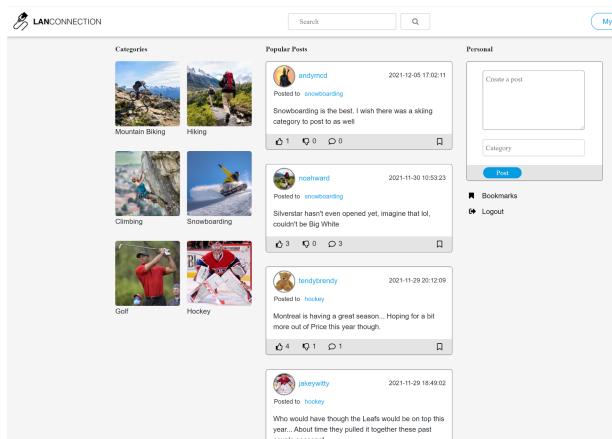
(Can use username=jd and password=password, or create a new account for this section)

6. Login/Signup Page



From the navigation bar you can access the login/signup page. Here you can either sign in using your username or email with your password, or sign up by providing all your info. You can choose to provide a profile picture as well.

7. Home Page with Updated Navigation Bar



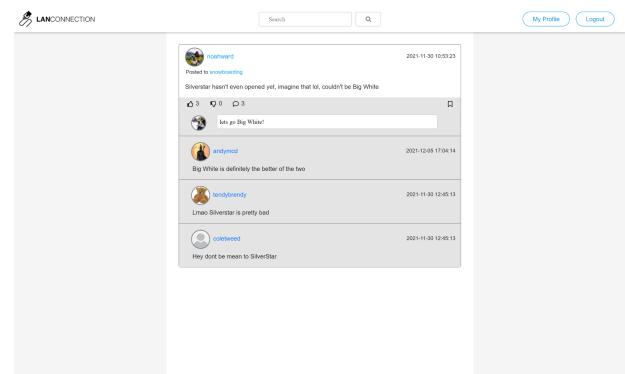
When logged in, the navigation bar is updated to have a link to your profile page and logout link instead of the login link.

8. Create New Post

The screenshot shows the LANconnection profile page for a user named 'andyred'. On the left, there's a large text area containing the text 'I got some good runs in today!'. Below it is a category selection box labeled 'Snowboarding'. At the bottom is a blue 'Post' button. On the right side, there are links for 'Bookmarks' and 'Logout'.

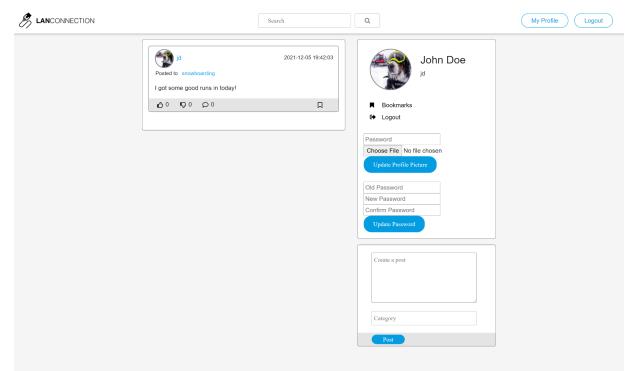
When logged in, you can create new posts in any of the categories from either the home page or your profile page.

9. Comment on Existing Post



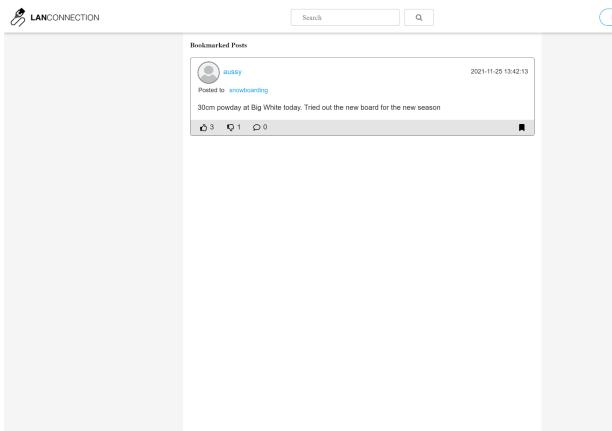
When viewing a post's comments while logged in, you can also add your own comment. And you can like or dislike posts and this will be saved.

10. View your Profile Page



When viewing your own profile page, you can change your profile picture or your password by completing either of the forms on the right hand side of the page. When the page loads without any type of error message you know the change was done successfully.

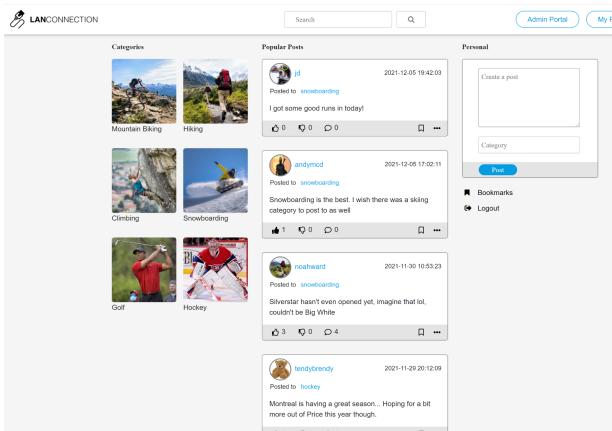
11. View Bookmarked Posts



At the bottom right of all posts there is a bookmark icon. A logged in user can click on this icon to add a post to their bookmarks. Then by following the Bookmarks link, on either the home page or your profile page, can view all their bookmarked posts.

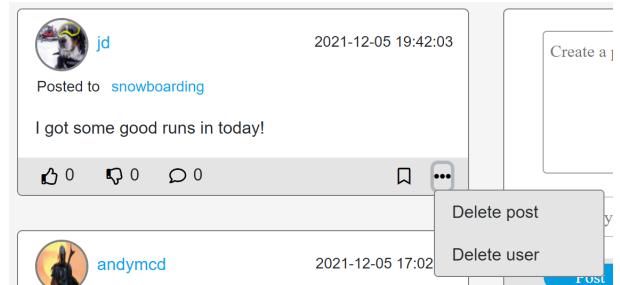
Section 3: Logged-In as Admin
(Use username=andymcd and password=password to view administrator functions)

12. Home page with updated nav bar



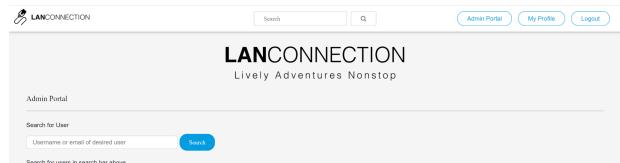
When an admin is logged in, the navigation bar is updated with a link to the Admin Portal.

13. Admin Tools on all Posts



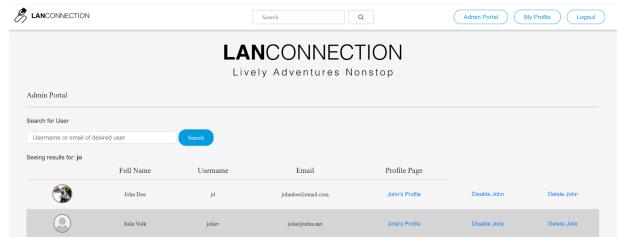
All posts also contain a new icon in the bottom right that allows an admin to delete the post, or the user who created the post. A confirmation is required for both of these actions.

14. Admin Portal



Once in the Admin Portal, an admin can search for users via their username, email, and posts.

15. Search User Results



From the Admin Portal, an admin can enable and disable users (which stops them from being able to login), can view their profile, and can permanently delete them.