University of Arkansas – CSCE Department
Capstone II –Final Report– Spring2020

NWA Hunter Jumper Association Dynamic Website

By:

Caleb Board, Adam Dover, Corbin Martin, Adam Hardman, Fadi Tommalieh, John Wolf.

Abstract

The Northwest Arkansas Hunter Jumper Association (NWAHJA) is a non-profit organization for English style riding. NWAHJA provides horse and riders, of all levels and ages, a venue to learn traditions, skills, and sportsmanship. They manage sanctioned horse shows where riders can compete in a professionally managed and judged events.

The Northwest Arkansas Hunter Jumper Association (NWAHJA) needs a dynamic website that help simplify managing their non-profit organization. Their situation requires an intuitive website that can receive input from NWAHJA members, equine trainers, administrators and in some cases even the general public. The input will be stored in a database so that the website can generate formatted web pages to display the data in a desired manner viewable only by the desired audience. The website will also serve as a cloud storage for monthly and annual reports.

Contents

1.0 Problem

The NWAHJA website project will have to be designed so it can be administered with limited technical website skills because every two years half of the board members who manage the program change out for a fresh set of volunteers. This will require the website to be designed so there are data input pages only accessible by approved roles. It will require online help FAQs to assist the website users based on their role. The website will dynamically generate webpages from content stored in a database that can be seen by approved roles. Roles will consist of website admin, board members, NWAHJA trainers, association members, and the general public.

There are different event types that NWAHJA manage, each have some unique requirements that make just about any typical open source event management software fail to meet the requirements without complicated modifications.

2.0 Objective

The objective of this project is to build a dynamic and intuitive website that can complete many core functions required to manage memberships, events, sponsorships, schedules, photo gallery, classified ads, the board member cloud repository for reports and financial statements, and online payment processing. By building this website, it will simplify the annual changing of board members, by providing a platform for the basic core functions of NWAHJA so it can continue to serve the area English riders. It will provide a stable platform that will simplify the requirements of future board members so they can spend more time working with the horses and riders and less time managing the business to make it possible.

3.0 Background

3.1 Key Concepts

Building a dynamic website to meet the needs of NWAHJA will require a scope of requirements, a front-end accessible via the internet with custom views by user role. The database back end will consist of a database to store and secure the data by role and custom logic to feed the data required for each view by role.

The scope of requirements will require interview skills and learning the business processes NWAHJA would like to occur online. It takes patience and a lot of discussion to complete a good scope of requirements. This process must get due diligence because it can make or break any software implementation project.

Front-end views on the website must render data in a format that displays properly on a PC or a smartphone. We will utilize the popular general-purpose scripting language PHP, HTML5 and JavaScript with the open-source CSS framework Bootstrap to accomplish this. Bootstrap contains CSS and JavaScript-based design templates which we will be able to modify to get a custom look that will meet NWAHJA's needs. The front-end will render data and graphics from a database driven feed so the website will not require manual webpage changes.

The backend programming will function in PHP, JavaScript, and other languages if needed, with all data being stored in MySQL database. User authentication will be required to identify non-public roles so user can only access data their role is authorized.

3.2 Related Work

NWAHJA's current website is mostly a static website written in HTML, which requires a webmaster with coding skills to update the website and data. They have membership signup and banquet signup which is interfaced with a MySQL database to collect member profile data and banquet data. The current membership signup and banquet signup is written in PHP 5.6 and is currently non-functional because of the decremented code.

The NWAHJA website currently has a Piwigo photo gallery and would like it integrated into the new implementation.

4.0 Design

4.1 NWAHJA Website Requirements

- User authentication will be necessary for access to non-public data.
- Automated password reset option will be required. New temporary passwords will be emailed to email address on the users account.
- A table of role types will consist of:
    - Association membership
    - Trainer
    - Riders (Associated with a membership.)
    - Board Member (By title such as President, Treasurer, etc.)
    - Sponsor Membership (By sponsorship level such as Show, Event, Medal, Advertiser, etc.)
    - Webmaster (Full administrative rights)
- Event types will include:
    - Horse shows (This will contain classes as sub-events and stabling options.)
    - Banquets (This will contain guest information.)
    - Training clinics (This will contain classes as sub-events and stabling options.)
- User profile will include a user role that will be established by user membership status or by a role of authority.
    - A list will be defined by read/write/delete rules for each role.
    -
- All web pages will be dynamically generated from data pulled from a database. It should refresh displayed content each time the page is opened or refreshed.
- All web pages will have a list of key web pages organized by Main Menu, Reference and Applications, For Our Members, For Board Members, About NWAHJA, and Social Media Links. These lists will pull from a database table so it can easily be updated by editing the contents of the data table.
- All pages will have a header containing logo, title, mailing address, donate button and menu ribbon of page links.
- All links on pages will only appear based on user role. This will make the header menu ribbon and footer of a page only show links that are available to the user by user role.

- The **Home** page will consist of a changing Photo Presentation and Mission Statement in the sub-header section. Other sections will be displayed as listed here: Notices containing near future events and important announcements, Sponsorships (Show and Event level only), Volunteering for NWAHJA section, Membership Application section, Scholarship Application section, and MyRider Application section.
- The **Events** page will consist of all events for current year and previous year. Events will be separated by event types in chronological order. Each event will have its own section so hyperlinks can be used to directly go to that section. Each group by event type will have a sub-header section containing a list of links within that event group. Past events are never deleted. They will be archived and there will be a modal to present web events older than the previous year's events.
- The **Sponsor** page will consist of a list of sponsors separated by sponsorship level. Sponsorship level determines the size and amount of space dedicated. The header of the sponsor page will consist of information about becoming a sponsor section.
- The **Member Barns** page will consist of a list of member barns by title in the sub-header section. Each barn will have a section to display barn logos, barn address and phone, trainers for the barn, barns webpage link, and an active map showing location.
- The **Contact** page will be a page with contact information and a message form where anyone can send a message to NWAHJA.
- The **About** page will consist of Mission Story, Board member collage with each board members contact information and a Jr. Board members collage with photo, name, and title only.
- The **Volunteering (still under construction)** page will have a splash screen explaining volunteering opportunities, a link to a report of volunteer hours to date by volunteer name, and a data entry form for volunteer hours to be entered by a board member.
- A **Blog (still under construction)** page where only board members, trainers, members, and sponsors can contribute stories to the blog. Board members can remove anything added to the blog.
- A **Piwigo Photo Gallery** page. NWAHJA already have this page and would like it to be integrated into the website.
- All **Create, Review, Update, and Delete (CRUD)** (many still under construction) data forms will have access controls by role for all data required for the website.
- Board Member cloud storage for contracts, financial statements, insurance policies, board reports, board meeting minutes, etc. This will have input screens to upload files and should be secure from all accept board members and webmaster. Files will all be assigned a category and program year. The categories list should be dynamic so new categories can be added as required by board member. Only a webmaster and Board President and Secretary will be able to delete files.
- A custom **Online Store (still under construction)** so users can acquire membership, register for events, and purchase NWAHJA merchandise with a shopping cart system and payment processing through PayPal.
    - Note only trainers will be able to add riders to horse show events.
- A **Classified Ads (still under construction)** page where all users can advertise equine related equipment, clothing, horse tack, etc. Data input form so the user can input the optional photos, item title, description, price, and contact information. But the ad will not display on the classified ads page until approved by a board member. Ads will discontinue after 90 days unless user acknowledges the merchandise is still available. User will have the ability to edit/drop/acknowledge the add. No sales will be handled by NWAHJA for classified ads.

4.2 High Level Architecture

In high level standards, we will design and develop a dynamic website by using the following programming languages.

PHP: Stands for (Hypertext preprocessor), uses server-side scripting language which runs on the web server. Is used to develop static or dynamic websites or web applications.

PHP Framework: We chose to use Laravel as our 3rd revision of the PHP framework of choice. Laravel is a PHP framework, designed to utilize a model, view, controller (MVC) technique to control the communications between the web front end and back end. (Laravel, 2020)

JavaScript: Object-oriented programming language that uses client-side functions. Used to create responsive and interactive elements for the web pages.

MySQL: Database language used to store information about user member, and it’s used to communicate with Database and it is standard language for relational database management system, and it can perform tasks such as update and retrieve data from database. (Tutorialspoint, 2019)

Bootstrap: A framework to help the user to design websites faster, it includes HTML and CSS based design templates for typography, forms, buttons, tables, and navigation. (Bootstrap, 2020)

Glyphicons: Icon fonts that can be used in web projects to provide a uniform user experience across all pages.

Hosting: NWAHJA uses Bluehost which is a well-known hosting platform. For hosting the website and for purchasing domain names. Used because NWAHJA has contract with them.

PayPal: is an online financial service that allows the user to pay for items using a secure internet account, the user can simply add their bank account, and whenever the user wants to pay using PayPal, they can choose which of their cards or accounts they want to pay with. Used because NWAHJA has contract with them.

Piwigo: It is an open source photo gallery software for websites which NWAHJA currently has implemented it and wants to continue to use.

4.3 Risks

Risk	Risk Reduction
If the website is not completely dynamic, it would still require some updates to be hard-coded. This is a risk because NWAHJA does not always have a qualified volunteer webmaster.	Generate all website pages from data pulled from a database. There will be forms for all data entry to enter data into the database.
The website needs to be intuitive enough so there is a small learning curve for users and administrators. Remember the NWAHJA administration are volunteers for two-year terms where 50% of the administration change annually.	Keep it simple and intuitive.
Data confidentiality.	The website will have to be designed so any confidential data is protected from unauthorized users. Role based rules will control which users can access/create/edit/delete data.
Data integrity.	Input/Edit forms will assist with accuracy with data type, error checking, and have controlled access by user role.
Data availability.	The backend controls will need to be thoroughly tested to ensure MySQL queries pull the required data for the user. If a query is done incorrectly it can cause incorrect information to be displayed.
Choosing an appropriate platform.	Ensure the programming language, the database, the language framework, and other components of the platform are current. Deprecated components can lead to critical losses of time and work.

4.4 Sprints/Tasks

Task 1 - Define development environment for the website. This includes the git setup, deployment and development on local machines, and the tasks planning provider. It is currently planned to use two git repositories, one for the database schema and another for the PHP combined frontend/backend.

Task 2 - Migrate existing project to a unified layout/framework. The existing work done by John Wolf needs to be reorganized and designed around a single framework. This framework is still up for discussion, either creating our own framework or using something like CakePHP or CodeIgniter. *(This task caused several weeks of loss work due to starting with CodeIgniter and realizing it was deprecated. We then switched to CakePHP and found it not stable. We then switched to Laravel PHP framework and it has proven to be a successful framework.)*

Task 3 - Design basic UI for user facing pages. This should first be mocked with wireframes and then agreed upon by the team members. The created UI needs to be usable for programmatically interacting with it and displaying database content. It will involve only the pages that logged in users can interact with. This includes the membership purchasing, the banquet purchasing, the account membership, and other additional pages as defined by John Wolf. It will also include account creation to access these pages.

Task 4 - Integrate backend with user facing pages. The created pages need to have their functionality and display components integrated with mock data in the database. All form submissions should be created, and validations made in this stage. The database schema will also need to be implemented.

Task 5 - Test user facing pages for bugs/vulnerabilities. All submission pages need to be tested against with bad data and injection phrases to verify they are properly validated. This should be repeated using direct posts to the endpoints to verify that validation is done on backend, not just client side. All display components need to be checked to make sure they only display information that the associated user can see. The backend itself needs to be checked that only the endpoints tested above are available.

Task 6 - Design basic UI for Admin Pages. This should first be mocked with wireframes and then agreed upon by the team members. The created UI needs to be usable for programmatically interacting with it and displaying database content. It will involve only the pages that administrators interact with to maintain the website. This includes confirmations, member and invoice viewing, event creation, and other additional pages as defined by John Wolf.

Task 7 - Integrate backend with admin pages. The created pages need to have their functionality and display components integrated with mock data in the database. All form submissions should be created, and validations made in this stage. The database schema will also need to be implemented.

Task 8 - Test admin pages for bugs/vulnerabilities. All submission pages need to be tested against with bad data and injection phrases to verify they are properly validated. This should be repeated using direct posts to the endpoints to verify that validation is done on backend, not just client side. All display components need to be checked to make sure they only display information that the associated user can see. The backend itself needs to be checked that only the endpoints tested above are available.

Task 9 - Design basic UI for front-facing pages. This should first be mocked with wireframes and then agreed upon by the team members. The created UI needs to be usable for programmatically interacting with it and displaying database content. It will involve all other pages not already made. Including the home page, upcoming events, the table of contents, directory, and other additional pages as defined by John Wolf. It will also integrate with the user and admin facing pages.

Task 10 - Integrate backend with front-facing pages. The created pages need to have their functionality and display components integrated with mock data in the database. All form submissions should be created, and validations made in this stage. The database schema will also need to be implemented.

Task 11 - Cleanup and unify backend. All code should undergo a code review and cleanup stage to minimize technical debt. Documentation for the codebase should be created before the review.

Task 12 (under construction) - Full test on all implemented code. All components need to be tested again. All endpoints need to be checked against bad input and intentional attacks. Overall functionality needs to be checked and to make sure the user state/session stays constant and only destroys when expected.

Task 13 (under construction) - Redesign and finalize UI. Do a final touchup pass on the UI to make it consistent throughout all the pages.

Task 14 (under construction) - Integrate invoicing with PayPal Sandbox. PayPal offers a testing environment for payments. This should be used with all invoice features and tested thoroughly.

Task 15 (under construction) - Do UX testing with real users. Ask real users to complete tasks on the website and take their suggestions. Anytime they struggle to complete a task take note.

Task 16 (under construction) - Implement requested UX changes and repeat task 15. Problem areas and suggestions should be reviewed and if agreed that a better solution could be implemented then do so. Once these changes are implemented ask more real users to test and continue doing so until no more major problem areas are found.

Task 17 (coming soon) - Get approval of implementation from NWAHJA Board. The website needs to be approved by the board. If they block certain features or want more, then this can spread into many subtasks of implementing those changes.

Task 18 (under construction) - Switch payment systems to live implementation. Switch PayPal over to their live API over the sandbox.

Task 19 (coming soon) - Migrate existing data. All data from the existing website needs to be brought into our new database. This includes previous events, previous members, previous invoices, etc.

Task 20 (coming soon) - Go to production. Host the website on NWAHJA's Bluehost and migrate the database from development into Bluehost.

4.5 Schedule

Tasks	Estimated Time	Estimated Start Date	Estimated End Date	Assigned
1. Define Dev Env	2 weeks	12/01	12/14	Corbin, Adam H, John
2. Migrate	4 weeks	12/15	01/12	ALL
3. UI User	2 days	01/13	01/15	Fadi & John
4. Backend User	1 week	01/16	01/23	ALL
5. Test User	1 day	01/24	01/25	ALL
6. UI Admin	2 days	01/26	01/28	Fadi & John
7. Backend Admin	1 week	01/29	02/05	ALL
8. Test Admin	1 day	02/06	02/07	ALL
9. UI Front	2 days	02/08	02/10	Fadi & John
10. Backend Front	3 days	02/11	02/14	ALL
11. Unify/Cleanup	1 week	02/15	02/22	ALL
12. Test All	3 days	not started	02/26	ALL
13. Finalize UI	2 days	not started	02/29	ALL
14. Invoicing	1 week	not started	03/07	John, Adam H, Corbin
15. UX Testing	1 week	not started	03/15	Fadi & John
16. UX Changes	2 days	not started	03/18	ALL
17. Board Approval	1 week	not started	03/26	John
18. Live Payments	1 day	not started	03/28	John
19. Data Migration	1 week	not started	04/05	John & Adam D
20. Production	1 week	not started	04/13	John

4.6 Deliverables

The project is around **60% complete** as a result of production coming to a near halt due the COVID-19 epidemic. Johnny Wolf will continue to work on the project through the summer to see it completed and implemented.

When the project is complete, we deliver the following:

- An implementation document that will explain how to implement the database, the website, and modifications that required in the code for hosting environment.
- Database creation code to implement it the database.
- Website code base.

5.0 Key Personnel

Caleb Board - Caleb Board is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has experience in group programming from taking Software Engineering, where a team had to create an app demo over the course of the semester.

Fadi Tommalieh - Fadi Tommalieh is a senior Computer science major in Computer Science and Computer Engineering Department at the University of Arkansas. He has experience with building and programming websites using HTML, CSS, JavaScript, PHP. He also can program in C++ and JAVA, and can work with teamwork, can build Oracle virtual machine server using VM, and has had an internship with Whyte Spyder to create Email Marketing and to publish using MailChimp.

Johnny Wolf – Wolf is a non-traditional senior completing a B.S degree in Computer Science with the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed all required CSCE courses except for Capstone. He has experience, relative to this project, in PHP, CSS, HTML, HTML5, JavaScript, and MySQL. He also has worked in a volunteer role with the NWAHJA board members to gain a strong understanding of NWAHJA's website needs. He is the current volunteer webmaster for NWAHJA.ORG and built the current website and membership and banquet signup process. Wolf retired from Wal-Mart Stores Inc. after 30 years. 27 years were in leadership roles. He retired to pursue his interests in computer science.

Adam Dover - Dover is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Database Management Systems, Software Engineering, and is currently taking Algorithms. He has relevant experience in JavaScript, HTML, and MySQL.

Corbin Martin - Martin is a senior CSCE Major and Math Minor at the University of Arkansas. He has worked for the past 4 years as a Technical Architect Intern at Tyson Foods Inc. with experience in AWS, Node.JS, C#, and SAP HANA. He has completed Database Management Systems, Software Engineering, Information Security, Computer Networks, and Algorithms. He has created a PHP based website for a private company in the past.

Adam Hardman - Hardman is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. Hardman has spent two summers interning for Cerner Corporation developing patient healthcare and management systems. He has experience in C# (.NET), Java, Python, and MySQL database administration.

6.0 Facilities and Equipment

For the project, we will not require any facilities except spaces to work and meet, nor will we require any equipment beyond our computers and general office supplies. The site will be hosted with Bluehost, so we will not require any servers of our own, and we will use Git or a derivative for version control. We have some shared group documents stored in Google Drive, so we do not need a separate space to store reports or planning documents. We primarily use GroupMe to schedule meetings and communicate, so we do not require any special equipment for those purposes.

References

Asa Hutchinson, G. o. (2020, March 30). *Covid-19 Proclamation*. Retrieved from Arkansas Govenor - Asa Hutchinson: https://governor.arkansas.gov/images/uploads/executiveOrders/EO_20-10._.pdf

Bootstrap. (2020, January 8). *getBootstrap*. Retrieved from Bootstrap: https://getbootstrap.com/

Laravel. (2020, February 14). *Laravel*. Retrieved from Larvel.com: https://laravel.com/docs/6.x

Sterling Systems & Controls, Inc. (2020, March 30). *Food Ingredient Batching*. Retrieved from Sterling Systems & Controls, Inc.: https://www.sterlingcontrols.com/applications/food-batching-systems/

Tutorialspoint. (2019, December 16). *MySQL - Learn MySQL*. Retrieved from TutorialsPoint: https://www.tutorialspoint.com/mysql/index.htm