Optimizing the Research Inventory Database for the Hudson/Mohawk River Watershed Project

Daniel Farkas, Abdellah Chemrah, Kishan Patel, Chirag Shah, Krishna Chakka, and Pranav Narvankar
Seidenberg School of CSIS, Pace University, Pleasantville, New York
{dfarkas, ac37339n, kp31080n, cs84942n, pn36097n}@pace.edu

***Abstract*— With the expansion and evolution of digital technology, the amount of data that people deal with increases significantly. The use of an automated and computerized approach to store and organize vast amount of information has become a necessity. Relying on traditional methods such as paper or simple spreadsheets is becoming impractical and obsolete. It is time consuming and costly to handle large amounts of data manually, especially for businesses. Nowadays, billions of dollars of transactions are conducted over the internet. What facilitates this process and makes it feasible is the use of online systems that store and process data rapidly. These data are stored in online repositories, or databases, and are retrieved instantly when needed. This entire process is convenient for both the provider and the consumer. The purpose of this project is to optimize and improve a previously developed inventory system for research papers. This system is in the form of a web application, or an online repository, that is accessible over the internet via an interactive interface.**

Keywords: PHP, MySQL, database system, graphical user interface, web application development, MVC design pattern, client-server architecture.

I. INTRODUCTION

The study focused on developing a web application that will hold all the information about various research projects conducted, or being conducted, in the Hudson Valley Mohawk River (HVMR) watershed, in Upstate, New York. This online database serves as an electronic repository for storing and managing research papers and various details about them. These details include information about the researchers and their studies (e.g., category, location, date, publication status, and citation). The reports about the projects are stored in a dynamic database that is manageable via a graphical user interface (GUI) and is made available on the internet [1].

A. System architecture and design pattern

The web application is composed of three major parts that are based on the VMC (View-Model-Controller) design pattern – a separation of concern model used in software and systems development [2]. First, the front-end, or the *View* component of VMC model, which is the GUI that the user sees initially when the application is accessed, usually through a web browser (Figure 1). With this dynamic GUI, the user can read, write and update the content of the database. Secondly, the *Model* component of the VMC design, which is responsible for handling the requests made thru the GUI by the user. A computer server is used to respond and interpret these actions and execute the commands to affect the back-end of the application. Thirdly, this back-end component, or the *Controller* components of the VMC, represents the database itself where the data are stored and arranged into tables of rows and columns [3].

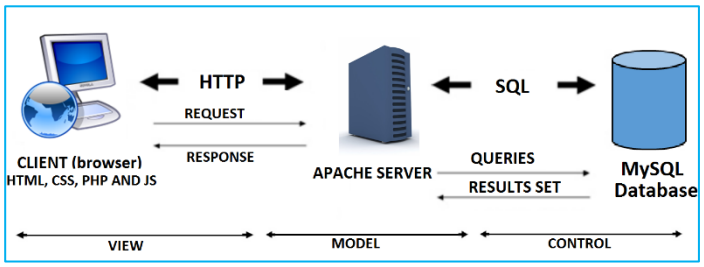

Figure 1. System Architecture

B. The users of the system

The web application is intended to be used by four types of users, with varying tasks, access rights and privileges. First, the *super administrator* who has full access to the system and can create users and assign levels of access and rights. Second, there is the *regular administrator* who has limited access and capabilities. Their main tasks include inserting data, updating and deleting them. Third, the members, who are the authors or researchers, will have their accounts and be able to write, delete and update their own papers and reports only. Finally, the public user, who does not need to have an account to search through the repository, view, save and print reports.

C. Technology and development tools

To build the system, various technologies were

used for different models of the application. There are many languages and platforms available for web design and development. One of the widely used building environments comes bundled in a suite called XAMPP, which combines MySQL database management system, an Apache Sever, and the PHP Scripting Language interpreters. More details about these components will be discussed later.

The front-end or the GUI was designed using HTML5 and CSS6, for the structure, layout and styles of the web pages. For the interactivity of these pages with the user, two main scripting languages, PHP and JavaScript, were embedded into HTML5 to carry instructions and queries to the back-end of the application.

PHP was originally called Personal Home Page and was used for basic programming of web pages. Since then it has evolved into a more advanced back-end scripting language and was renamed PHP: Hypertext Preprocessor. It is contained in the XAMPP suite and is available free of charge under general public license from Apache and Friends community [4]. This platform is compatible with different operating systems (OS) such as Linux, Window, and Macintouch. PHP has various built-in features that work well with web applications. This helps developers use functions like GET and POST, URLs and HTML easily [5]. Since PHP is an open source language, it has a large number of libraries and extensions that add more functions to it [6].

In addition to PHP, XAMPP also comes with different databases, their management tools, and consoles. One of these databases, which was the choice for this project, is MySQL, an open-source, cross-platform Relational Database Management System (RDBMS). It comes in two different editions: MySQL Community Server, the one used for this project, and Enterprise Server. The RDBMS is equipped with a GUI console that enables users to create and manipulate data stored in the online repositories [7]."

Using MySQL RDBMS, an admin can create and manipulate the database. Most commercial RDBMSs use the Structured Query Language (SQL) as a programming language to build databases and define queries to interact with them. Therefore, MySQL was an adequate option for web applications; it is flexible and easily used to manage databases.

The RDBMS is the intermediary between MySQL and the GUI console. Besides, the RDBMS facilitates the dynamic information storage and retrieval, eliminates data redundancy thorough normalization, and improves the consistency to increase the efficiency of the database's search and performance [8]. The console that provides the GUI to manage and create databases using MYSQL is called phpMyAdmin [9]. Operations such as managing databases, columns, relations, indexes, and users, are also performed via this interface. However, SQL statements can still be entered manually and executed directly on phpMyAdmin console [10].

Since there is a large community for PHP developers, a variety of support can be obtained from online forums and blogs. But despite the popularity of PHP, it has some security issues, such as SQL Injections and other harmful, malicious inputs. These concerns will be addressed when creating, troubleshooting the database, and coding the web pages. Some of the solutions to circumvent these problems are: input validation, data sanitization, and string escaping, as discussed towards the end of this report.

II. BACKGROUND

The Environmental Consortium of Colleges & Universities (ECCU) does numerous environmental researches in The HVMR region, which consists of one fourth of the larger Hudson River basin in Upstate, New York. The ECCU needed an online repository system to store its research discoveries and make them accessible for electronic retrieval by different entities such as Pace University and other organizations. In the past, the ECCU utilized unstructured spreadsheets to store data; that made the latter unorganized and difficult to search and sort. Fortunately, the online system is used to store and manage all the data about the researches done in that area, using a web application containing a structured online database.

This project was started previously by a team of Pace University students who detailed their work in a published technical paper entitled as "A Research Inventory Database for the Hudson/Mohawk River Watershed Project" [11]. The purpose of this system is to incorporate all the information obtained from the researches and make them available and easily searchable using precise criteria. To accomplish this, a web application was developed. It is constructed utilizing both PHP and MySQL that resides in an Apache Web Server [12]. PHP is utilized for the Server-Side Scripting of the web application, while MySQL DBMS is used to create a database in the form of interrelated tables that hold different types of data and make them available to retrieve, update and delete, as illustrated in Figure 1 previously [13].

The previous team had achieved the results by designing a simple and clear wireframe for the web, interface, which was later implemented using, HTML5 and CSS [14]. They laid out a schema for the MySQL DB with various relations between tables [15]. The prototype completed contains all these features in the form of a web

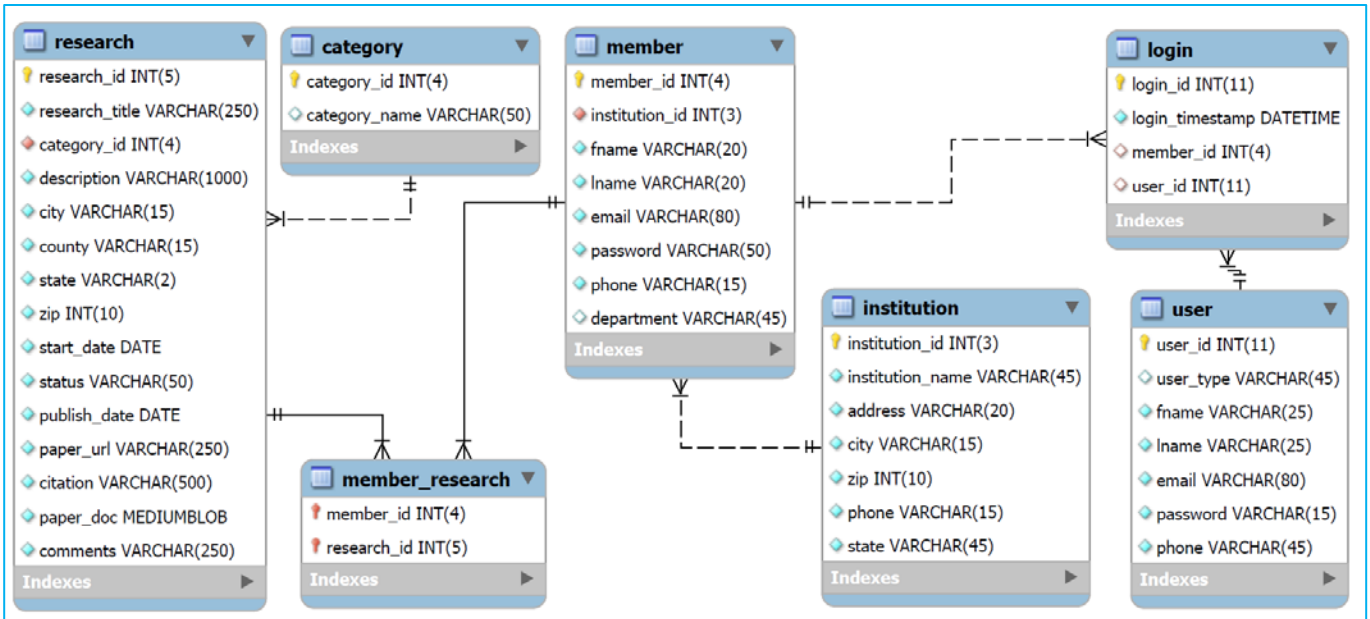

Figure 2. Revised Database Schema

application. Subsequently, our team became in charge of the project to improve, optimize, and extend the features of the system. The work achieved is summarized in the following segments.

A. Previous Development Cycle Overview

The application was improved, debugged and new functionalities were implemented as requested by the customer. These results were accomplished by fixing the existing issues, cleaning up the GUI and the search fields, and optimizing the back-end code.

Upon frequent physical meetings, collective effort and communication the system was successfully enhanced and deployed. To achieve this, various collaborative tools such as Email attachments, GitHub and Google Docs were used to share resources and work on them concurrently. Besides, instant group messaging such as WhatsApp and email services were used for communication between the members of the team. The rest of this section describes the results achieved and illustrates them through screenshots taken from the working system.

B. Bug Removal

As a first step in improving the system, the web application was run, the source code was debugged for apparent issues and malfunctions. To do so, the PHP scripts for each existing page were investigated and fixed before composing new ones. Additional pages and blocks of code were added to fix and incorporate more features as the project evolved.

C. Database Schema Improvement

Upon changing the fields of the search criteria in the GUI, adjustments to the database schema and tables had to be made, as shown below in Figure 2. The Schema shows the tables in the database and how they are interrelated, since MySQL was used as a RDBMS [16]. The relations between the tables have also been improved.

The schema was constantly checked to ensure the accuracy of the data types and minimize errors and security risks through input validation. A well-designed database ensures that the relations between its different entities are appropriate to avoid redundant and inconsistent data. Consequently, adjustments to the back-end code were handled accordingly.

D. Search Field Modification

In addition to fixing bugs and improving the script, the search fields of the GUI were modified as the customer requested.

The first improvement made was to the search fields as shown below in Table 1. Some fields were added and others were combined or omitted. The reason behind these changes was not only to make the search interface clear and easy to the user, but to also increase the efficiency and accuracy of the search.

E. The GUI Improvement

The modifications made to the search fields and database generated changes to the GUI, the front end of the web application. Based on these changes the layout of the GUI was improved and made easy to navigate. The GUI is the interaction of the user with the system. An unpleasant or complex GUI can make the user experience unfriendly and eventually leave the application within seconds [17]. Figure 3 shows the home page, which is the first interface that is displayed to the user. The user then opts to login with an existing account, register or simply select a search field to search for a specific information.

Table 1. Revised Advanced Search Fields

Field Name	*Description*
Research Title	Title of study or keywords
First Name	Author's first name
Last Name	Author's last name
Institution	College or university
Category	Category of research
Start Date	Start date of research
Publish Date	Publish date of research
Location	City, county, zip or state
Description	Brief project description
Status	Status of publication

In addition to the front-end work done, new features and improvements were implemented in the back-end. First, the search queries and the back-end code of database was optimized. Based on the keywords and search criteria that the user enters, the number of results found is returned as well as the content. Second, the user can either choose to display all the existing records in the database or enter proper keywords to perform the search operation on the existing records. Third, the admins, unlike regular users, can not only search the repository, but also add, update or delete reports from it. It is from this front end that the back end (MySQL DB) is reached and manipulated. The queries were optimized and improved since the previous system's implementation. Instead of showing result that do not relate to the search keywords, our queries return precisely what is searched for, i.e., records containing some or the whole phrase entered.

III. New System Requirements

For this cycle of the development, the client had outlined an extensive list of new functional requirements and adjustments to the system.

After the successful completion and implementation of the system, including all the customer requirements from the previous cycle, it was time to maintain it, enhance and build new features. These requirements consisted mainly of the following major key points: improving the overall database schema and relations, adding stronger admin control access, and redesigning the layout of the web application's GUI. The list below contains a summary of these adjustments and the tasks that were to be accomplished:

- Cleaning up/modifying the interface
- Allowing public access to queries
- Adding different admin levels access controls
- Showing multiline return from search
- For search results, having print, delete, & modify
- Including an add new report function all over
- Linking to the Consortium and Map
- Adding a menu for Reports, Category, Institution
- Enhancing import all/export all functions.
- Adding security features–hashing & data sanitization

In addition to this list, a detailed document of specific requirements and changes to each page of the application had been provided by the client. These features were implemented and documented throughout the project's cycle. The following segments describe these aspects.

A. *Admin Access Control*

For reasons of security and integrity of the application and its data, access to the sensitive modules of the system was limited to the person(s) in charge of maintaining the system and its content. After the previous implementation, the system did not hold any actual or sensitive data, it was publicly accessible and modifiable. Consequently, this made it unsecure and vulnerable. Implementing strong restrictions resolved this issue. The *Design* section of the paper covers the overall layout of the admin's console and its different functions.

B. *Database Redesign and Enhancement*

Some fields of the tables in the database were redundant or irrelevant. Getting rid of or rebuilding them improved the relationships within the database, hence, ensured data consistency and search efficiency. As the system was being developed, the database and queries were tested and optimized. As a result of these changes, the schema shown previously on Figure 2 was revised and redesigned to reflect the requirements the client had specified. Some of these adjustments included: removing certain fields from the tables, adding new ones, modifying datatypes, and creating/dropping whole tables. This gave the ability to advance search based on multiple criteria and fields as illustrated in the *Design* and Implementation sections.

C. *GUI Layout Improvement*

The client had specified the content and layout of each page in great detail. Understandably so, ameliorating the user interface and maintaining a consistent look throughout the application is essential in web and software development in general.

As examples of these specifics, the customer stressed the need to maintain a consistent look on every

page and enable access to the various functions from anywhere within the application. Functions such as manipulating reports, managing users, and updating records pertaining to the members of Consortium and institutions that are affiliated with them. In addition, adding edit, delete and print feature to each search result, for only the admins with the right access. Besides, two fields were added to the layout to indicate the start and publish dates of a research, to help the user narrow down the search results, and determine whether it is completed or ongoing.

These were just of few from a long list of requirements. The changes did not only affect the GUI, but also reflected on the database fields and schema as discussed before. Subsequently, a huge deal of coding, testing and debugging also took place in order for the three main components of our system to work jointly as a whole.

The next section illustrates some of these designs and layouts through actual screenshots from the application.

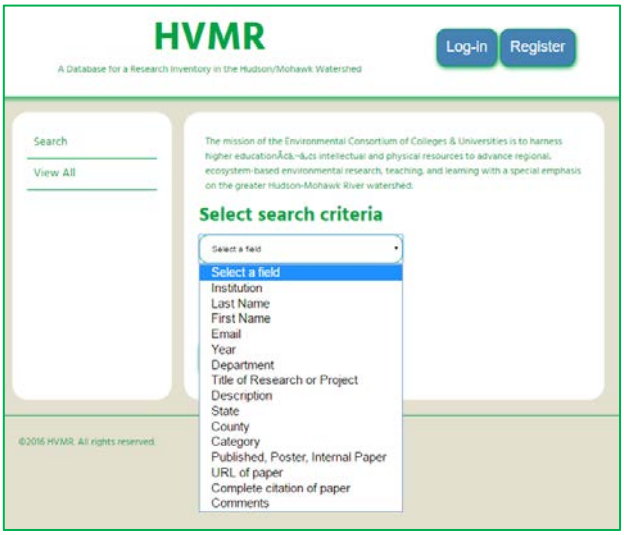

Figure 3. *GUI* of the home page

C. *Database Backup and Restore*

A database system can be compromised or can crash at any point, therefore, having a backup image of the data in a safe location is of great importance. In fact, the application was equipped with functions for this purpose. As shown in Figure 4, *import and export* buttons allow the administrator to both save the entire data in a preferred location and the give the ability to recover and restore them when needed. These functions were enhanced as discussed later in this paper.

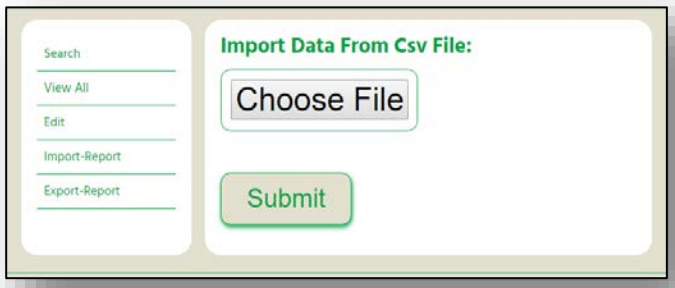

Figure 4: Import/Export function

IV. NEW SYSTEM DESIGN

A. *GUI/Home Page Layout*

As opposed to the previous layout, the new home page of the web application features a simpler, new GUI. Instead of having multiple search criteria drop down from a menu on the home page, the new look contains only one, main search field (Figure 5). This is convenient for a quick search based on keywords from a report or its title as users may not have all the details about a certain research or paper.

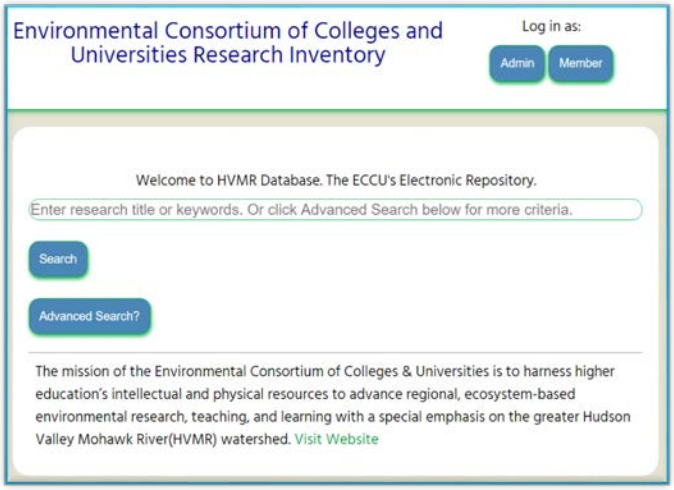

Figure 5. System's Home Page

B. *Advanced Search Page*

In order to run a precise and detailed query with several search criteria, the user can simply click on the Advanced Search button located on the home page. This opens up a new window with multiple search fields to choose from to narrow down the search results (Figure 10).

C. *Admin's Panel*

The previous implementation lacked administrative features and an access control mechanism. At that point, there were only two types of users: registered and public. While the public could only search,

save and print, the registered ones had equal access rights and the ability to add records and manipulate them. The aim of the new function was to have different users, with varying levels of access. First, the *super admin*, with full control over the application, can create, manage, and grant *regular admins* specific rights. These latter have limited access and can only add, delete, and update reports. Thirdly, there are the *members* (i.e., the scholars and researchers who join the ECCU). They are able to post their own reports and edit their entries only. Finally, the public user, who can only search, save and/or print. No account or membership is required to do so.

An administrator with total control over the system can oversee and manage the whole application through the admin's panel shown in Figure 8.

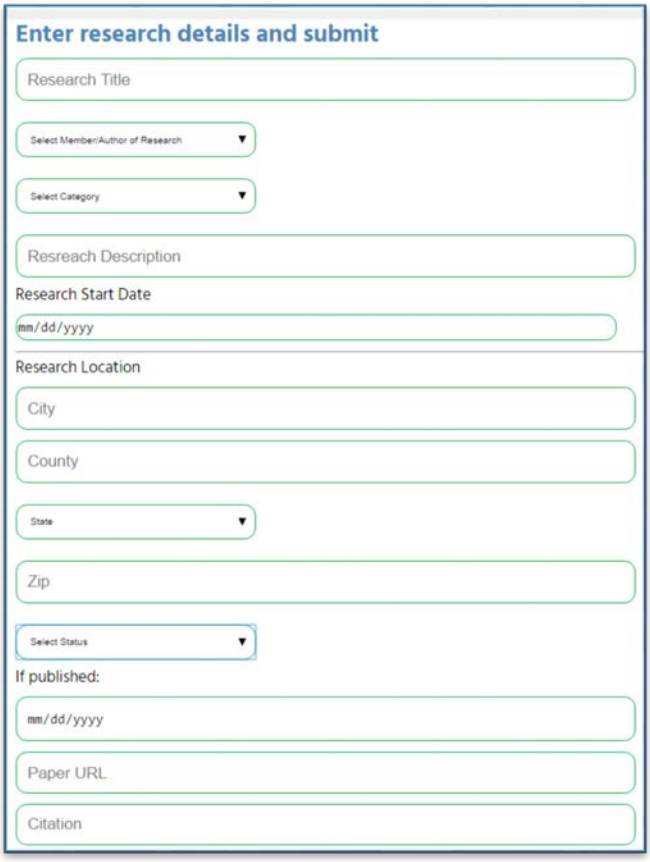

Figure 7. Report Entry Form

V. SYSTEM IMPLEMENTATION

The previously discussed functions were implemented and tested. They included: activating the admin's panel, inserting records to the database and manipulating them (via the same panel), and implementing the *Search* and *Advanced Search* functions

A. *Admin's Panel*

The navigation menu shown on Figure 8 contains the main control panel from an admin's point of view. Depending on the type of user logged in to the system, available functions on this menu vary based on access right levels. From this panel a user can enter a repot, edit exiting data, import/export the entire database, or manage the users. The last two functions are only visible to the super admin, who has a total control over the system.

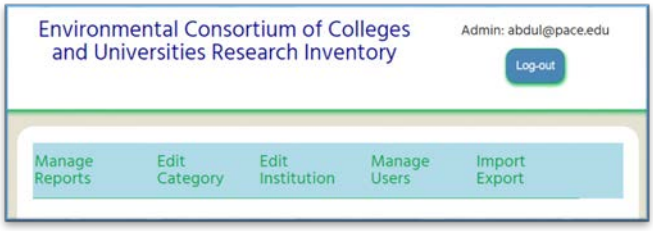

Figure 8. Admin's Control Panel

B. *Data Entry Form*

The first function made available to the admin users and the members is the ability to make entries to the database and update or delete the records as necessary from the GUI. The *New Entry* button of the control panel menu takes the user to a long form to fill up with details about a certain study or research, as shown in Figure 7. The information entered is then collected and stored properly in the right tables of the database. When the records are successfully inserted, a message is displayed, otherwise, the user is notified and a warning is issued. In addition, once the entry is made, the user has the options of viewing (Figure 9), updating, or deleting it, as shown in Figure 8.

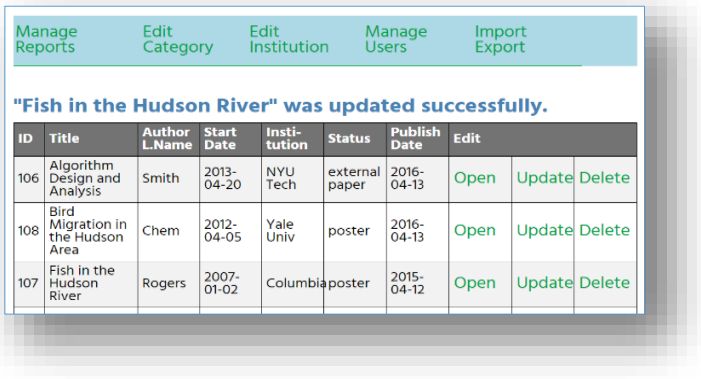

Figure 8. Report Management Page

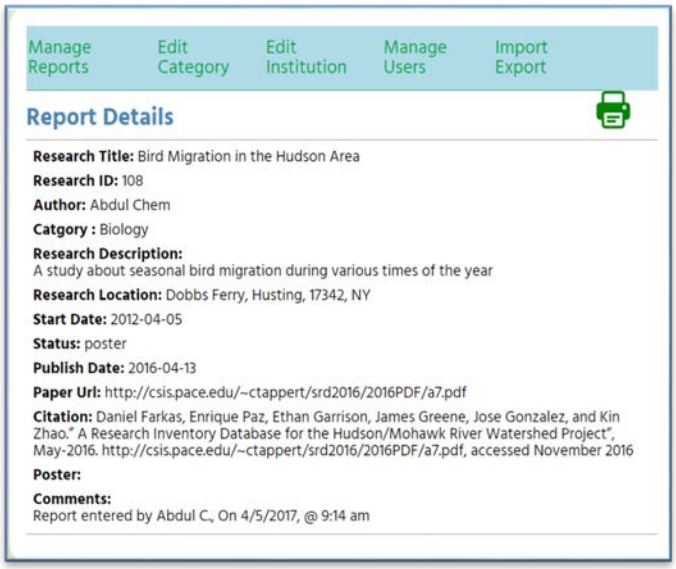

Figure 9. Report Details

C. *Basic and Advanced Searches*

After the data is gathered and sorted inside the repository, the other function implemented, and the most important one, provides the ability of searching through the database and pulling up the results desired, based on a number of precise criteria defined by the user.

The online application developed allows the user to run two types of searches: a regular one, where only *keywords* from the title, the paper, or its description are necessary to do a search. Additionally, there is an *advanced search* with numerous search fields (Figure 10) that enables the user to narrow down the results returned based on the specified criteria.

At least one field is required for this type of search. Using a long SQL query, the application delves into the database, sorts and retrieves the concise results. For instance, the user may look for a certain study, done by a specific person, from a known institution, in an exact category, and a location of their choosing. The user may even supply a time interval and/or the status of the study to narrow the results returned (Figure 11). Since a record with the specified criteria exists in the database, it was returned as a result. Changing only one criterion (e.g.; NY to NJ for the location), would not return any results, for the simple fact that the user explicitly wanted a study with all those criteria AND done in NY. Lastly, the *Advanced Search* function was inspired, in terms of the logic and design by Google's own advanced search, but the coding and implementation was done by our research team.

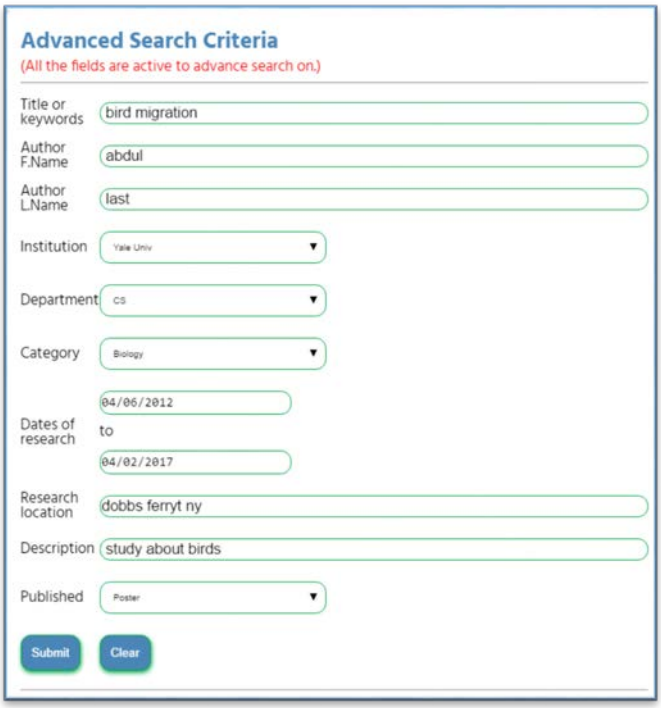

Figure 10. Advanced Search Form

Figure 11. Advanced Search Results

D. *Database Backup and Restore Functions*

Import

Inserting information into a database can be a tedious and a time-consuming process if done manually. Knowing that the customer currently stores data in Excel spreadsheet, an *Import* feature was built into the GUI to make this task fast and efficient. To do this,
the excel spreadsheet data needs to be saved as csv file, and then imported easily through the application's GUI, using the *ImportData* button, as illustrated in Figure 4[18].

Figure 4: Import/Export function

Export

Backups are an essential feature in any database system and without this function the data could be at risk. No system is 100% safe and reliable, hence, the need for saving copies of data regularly. Whether it is due to a system failure or a malicious act, a database can become dysfunctional, and if no backup is available, this could be a serious issue. An admin can export the entire records of the database simply by clicking *ExportData* button (Figure 4). The data are then downloaded as an Excel document [19].

In addition to this method of importing/exporting data through the GUI of the application -- which is a convenient and easy way for an admin with minimal knowledge of database management systems -- a second way of backing up the entire MySQL DB itself and schema is also detailed in the technical guide of the system. This method is for the advanced users and database admins; and it is for the purpose of making changes to the schema or the design at a later time if required.

E. Securing The System

Security is one of the main concerns in web and software development; building a totally secure and reliable application is almost impossible and is getting exponentially challenging, as the cyber-attacks increase in frequency and sophistication. One of the first steps to take in order to make a robust application that resists web attacks is to follow strong software design and development practices to reduce the risk of vulnerability. In fact, the prevention from such attacks starts with building a well-designed application, taking into consideration various factors that may leave an open door or a security hole to be exploited by hackers.

Various measures were taken while the system was being developed to ensure security and safety. The first step taken was implementing identify authentication and access controls. By doing so, only members or admins are able to log in to the system and have access to specific resources set by the super admin. In addition, passwords are hashed and salted first before being stored, rather their plain text version. This was accomplished using PHP's password_hash() function. Further, data sanitizing is important being inserting them into the database. This refers to escaping and ignoring special characters that can express malicious scripts to be injected in the forms that write to the database (filter_var and mysqli_real_escape_string are examples of PHP functions used for this purpose). Data validation is another important measure that was considered. This process adds more security by checking the user's input before it is stored in the database, to ensure the data types conforms to the rules. This prevents SQL Injections attacks where malicious code is embedded into an input field to retrieve info from the database. Finally, sessions and cookies were also considered to ensure the right content is accessed and displayed to the right users, based on their level of access and rights.

F. Implementation/Migration a Pace University Server

The system was developed and constantly tested on local machines using the XAMPP Development Suite, which simulates an actual real environment for developers. Once the system was completed, it was deployed to a Pace University server and is now accessible over the internet via the following link: http://vulcan.seidenberg.pace.edu/~f16-hmvr.

The process of migrating the system to the Pace environment was not as easy as anticipated. In fact, XAMPP is a complete package with PHP, MySQL, Apache Server and a GUI management consoles. Whereas the Pace's Vulcan server did not provide such tools, so the whole process of transferring the system and the database had to be done through command line.

VI. TESTING THE SYSTEM

The system was thoroughly tested before implementation to make sure all the models and components function as designed. The user experience is essential in testing and evaluating the final product. For this purpose, multiple users tried the system. As a results, when the submission of a report is completed, the records were properly inserted into the database and a message of the success is displayed. The user is then presented with the options of adding additional reports, updating or deleting them. At this point, a user may have several ways of searching through the systems reports including name, category, description, department, year…etc. The user can then search through a specific category of reports, view, or save them. After the search form is submitted, the user can immediately view the results, and if the results are not as desired, they can go back and submit a new search query. Finally, previously collected data in Excel sheets can be imported into the system with a click of a button. Similarly, the data can be exported for backup purposes through the GUI of the system as illustrated before.

VII. Evaluating the System

At this point, the system is functioning properly and it was migrated to the Pace University environment. The client can now import the current data that are stored in spreadsheets to the system. Second, a backup function is very important to include, to have a safe copy of data in case the system goes down for whatever reason. To accomplish this, the admin can easily export and backup data though the GUI as shown before.

The work done fulfilled the requirements the customer specified and beyond. The previous system was significantly improved and optimized. As technology evolves, the system should be tested and constantly maintained. As detailed in the user guide of the system, admins will have full control over the application. Besides, the source code and all the modules and components of the system were provided to the client for future work and maintenance of the system. As a matter of fact, software development is a cycle of ongoing maintenance and improvement. As this development cycle ended, a new one starts to keep the system secure and up-to-date. The client may request new requirements and functionalities to be build and implemented.

VIII. Conclusion

At the stage, the system is complete and the final product was delivered to the client and their users. After evaluating and testing it, the web application of database system was migrated to a web server. The screenshots shown above were only a few from many of the whole system. In addition, other technical features, not mentioned here, were also implemented and are visible online to the customer, the admins, and the public users, through the system's URL given above, on a Pace University server.

IX. References

[1] [1]B. J. Jansen , "The Graphical User Interface: An Introduction," The Graphical User Interface: An Introduction, 1998. [Online]. Available: https://faculty.ist.psu.edu/jjansen/academic/pubs/chi.html. [Accessed: 01-Apr-2017].

[2] "Model-View-Controller" Available: https://msdn.microsoft.com/enus/library/ff649643.aspx Accessed: March. 06, 2016.

[3] Purva Deshpande,"Sacramento Masters Projects-A data integration and olap application: university exploration tool for international students" Available: https://csus-dspace.calstate.edu/bitstream/handle/10211.3/181760/2016DeshpandePurva.pdf?sequence=3 Accessed: March 05, 2016.

[4] PHP, "What can PHP do? - Manual. "http://php.net/manual/en/intro-whatcando.php, Accessed September 2016.

[5] T. P. Group, "What can PHP do?," 2001. Available: http://php.net/manual/en/intro-whatcando.php. Accessed: Nov. 08, 2016.

[6] Advantages of PHP Programming," Web Design Library" Available:https://www.webdesign.org/web-programming/php/advantages-of-php-programming.21905.html. Accessed: March 05,2017

[8] "SQL RDBMS concepts," www.tutorialspoint.com, 2016. [Online]. Available: https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm. Accessed: Nov. 10, 2016.

[7] "MySQL PHP API :: 5.5.4 mysql_connect," MySQL. Available: https://dev.mysql.com/doc/apis-php/en/apis-php-function.mysql-connect.html. [Accessed: 01-Apr-2017].

[9] Apache, "Apache HTTP SERVER PROJECT" Available: https://httpd.apache.org/ ,accessed November 2016.

[10] p. contributors, "PhpMyAdmin," phpMyAdmin, 2003. Available: https://www.phpmyadmin.net. Accessed: Dec. 2016.

[11] Daniel Farkas, Enrique Paz, Ethan Garrison, James Greene, Jose Gonzalez, and Kin Zhao." A Research Inventory Database for the Hudson/Mohawk River Watershed Project", May-2016. http://csis.pace.edu/~ctappert/srd2016/2016PDF/a7.pdf, accessed November 2016

[12] [12]D. Group, "Essentials¶," About the Apache HTTP Server Project - The Apache HTTP Server Project. [Online]. Available: https://httpd.apache.org/ABOUT_APACHE.html. [Accessed: 02-Apr-2017]. [13] "PHP 5 introduction," Available:http://www.w3schools.com/php/php_intro.asp. Accessed: Nov. 08, 2016.

[13] "WordPress.org," Switching to PHP5 « WordPress Codex. [Online]. Available: https://codex.wordpress.org/Switching_to_PHP5. [Accessed: 25-Apr-2017].

[14] "5 good reasons why designers should code - treehouse Blog," in Learn, Treehouse Blog, 2010. Available:http://blog.teamtreehouse.com/5-good-reasons-why-designers-should-code. Accessed: Dec. 09, 2016.

[15] C. Heng, "What is MySQL? What is a database? What is SQL? (thesitewizard.com)," 2010. Available:https://www.thesitewizard.com/faqs/what-is-mysql-database.shtml. Accessed: Nov. 09, 2016.

[16] "Learn MySQL Fast, Easy and Fun.," MySQL Tutorial. [Online]. Available: http://www.mysqltutorial.org/. [Accessed: 09-Apr-2017].

[17] A. Hannemann et al., "Smashing magazine – for professional web designers and developers," Smashing Magazine. Available: https://www.smashingmagazine.com. Accessed: Dec. 06, 2016.

[18] A. Majid, "Import excel file data in MySQL database using PHP," in PHP, Abdullah Majid, 2015. Available: http://www.eggslab.net/import-excel-file-data-in-mysql-database-using-php/. Accessed: Dec. 07, 2016.

[19] jackgoddy 123, "Import excel file to php myadmin through file uploading using php," The SitePoint Forums, 2014. Available: https://www.sitepoint.com/community/t/import-excel-file-to-php-myadmin-through-file-uploading-using-php/38433. Accessed: Dec. 08, 2016.