Digital Object Identifier

HealthVault Hub: A Healthcare Database Management System

BRANDON SANDOVAL[1], NICHOLAS BRIDGES[2], AND CHRISTOPHER WITT.[3]

Corresponding author: Brandon Sandoval (e-mail: brs5s@mtmail.mtsu.edu).
Corresponding author: Nicholas Bridges (e-mail: njb4p@mtmail.mtsu.edu).
Corresponding author: Christopher Witt (e-mail: caw2ax@mtmail.mtsu.edu).

ABSTRACT One of the biggest industries in the world is Healthcare. It offers a multitude of different things regarding one's health: Primary Care, Hospital Care, and Health Information Technology. This project focuses on the aspect of Health Information Technology; furthermore, creating an online environment that suits the actions of different users. The Healthcare Database System that we have created offers the accessibility to a variety of medical history as well as other relevant information like billing or account history. The structure of the website follows the user to a main page. The user will then be able to select the type of account they are trying to access: patient or administrator. Based upon this, a prompt will come up asking for their username and password. When entering, they will be taken to the corresponding portal where they can access or update their records. The database will be written in MySQL and be made accessible through the use of JavaScript and especially PHP. The website itself will be composed of HTML and CSS coding to give a clear format for the user. The overall idea of this project is to provide a sufficient Healthcare system that resembles something from a major corporation. The purpose of this project is to learn about databases and the use of MySQL. Also, it provides a good understating for HTML, CSS, PHP, and JavaScript.

I. INTRODUCTION

HEALTHCARE is the most essential industry on the planet and is an important part of human life. With it being so important comes our project that centers its focus on the IT aspect of Hospitals. Health IT, which is a certain field of IT uses technology to enhance efficiency, accessibility of medical information, and minimizes human error has become a big role of modern day Healthcare. Our project specifically is focusing on address the evolving complexity of patient records by developing a sophisticated Healthcare Database System.

A system that which is designed to help either patients or medical staff with an easy to access online environment, in it would contain a comprehensive access to medical histories, billing information, and other miscellaneous data. In this online environment, our web-based system will direct users to a main page where they will be asked to identify as a medical staff or a patient and then be sent to the corresponding portal. This portal will allow the user to login through their credentials allowing patients to see their records and update basic information while allowing the medical staff to be able to navigate through records, updating of users' individual records and makes a user-friendly experience. This system's database is built on MySQL, a robust relational database management system, and on the front-end HTML, CSS, and JavaScript will be used to ensure clarity and interactivity to the web.

This project not only serves as a Healthcare Database System that represents a vital synergy of technological innovation and healthcare efficiency but as a priceless teaching resource. With our focus on MySQL, it will provide an engaging education in the field of databases. The front-end integration of HTML, CSS, and JavaScript also provides an understanding of web development. With a goal on delivering a healthcare system that resembles the sophistication of major corporations, emphasizing the data management and technology integration in light of the constantly changing healthcare environment. We investigate the intersection of technology and healthcare as we delve into the details of this project, illuminating the relationship between technology and man.

II. PROJECT OVERVIEW

THE project will very loosely follow similar designs of larger corporations but with a twist of our own creative design. The front will be supported by HTML, CSS, and JavaScript. It will feature a portal to choose if you are logging in as a patient or administrator. Given the correct credentials, it will then take the user to the correct website that allows them to access the information that is allowed to them. The use of HTML will make a good back bone for the front end code. With the help of the CSS file, we will be able to make the website look sophisticated and to our creativity's liking. The JavaScript portion of the code will allow us to grab the moving parts of the website (various information) and apply cool designs.

Moreover, the back end of our project will feature the use of MySQL for the database construction, and XAMPP using PHP to correctly link the front end to the back end. It will allow us to pull the information from the MySQL database and transfer it to our JavaScript, which will then display the information for the user to see. This will also let the user insert various different information into the database. The MySQL database will be constructed of various tables like Patient, Hospital, and Administration. Each of these tables will feature a different list of key components.

Furthermore, the overall website should work as follows: a user pulls up the website and puts the correct information in. Then, the user will chose what they want to do: view or change/add records. This will then command the back end to pull from the database. Given all this goes correctly, the user will then either get a confirmation or see the requested information.

III. METHODS

THROUGHOUT the building of our website, we experimented with various different things to give us the edge in creating the design later on. Of course, our project required various amounts of code already mentioned: HTML, JavaScript, PHP, CSS, SQL, and the use of XAMMP. I would say overall our biggest hurdle was the issue of connecting the information on both ends, but we found our rightful way with the use of XAMPP. Additionally, another hurdle we had was the understanding of the inter connected parts we had created. Furthermore, the beginning of the semester had all the information we needed for this project besides the crucial coding language PHP. Without the knowledge of PHP we have now, we would not of been able to complete this project. A very interesting thing that we have also learned with the knowledge that we have consumed is that our project leaned very heavily into JavaScript at the beginning of the semester. This was due to the fact that we thought we would be able to pull the information with PHP, and then we would be able to display the information with JavaScript. This turned out to not be true at all, which is not necessarily a bad thing. However, this lead to some headaches trying to connect and add functionality to the work we created in the beginning.

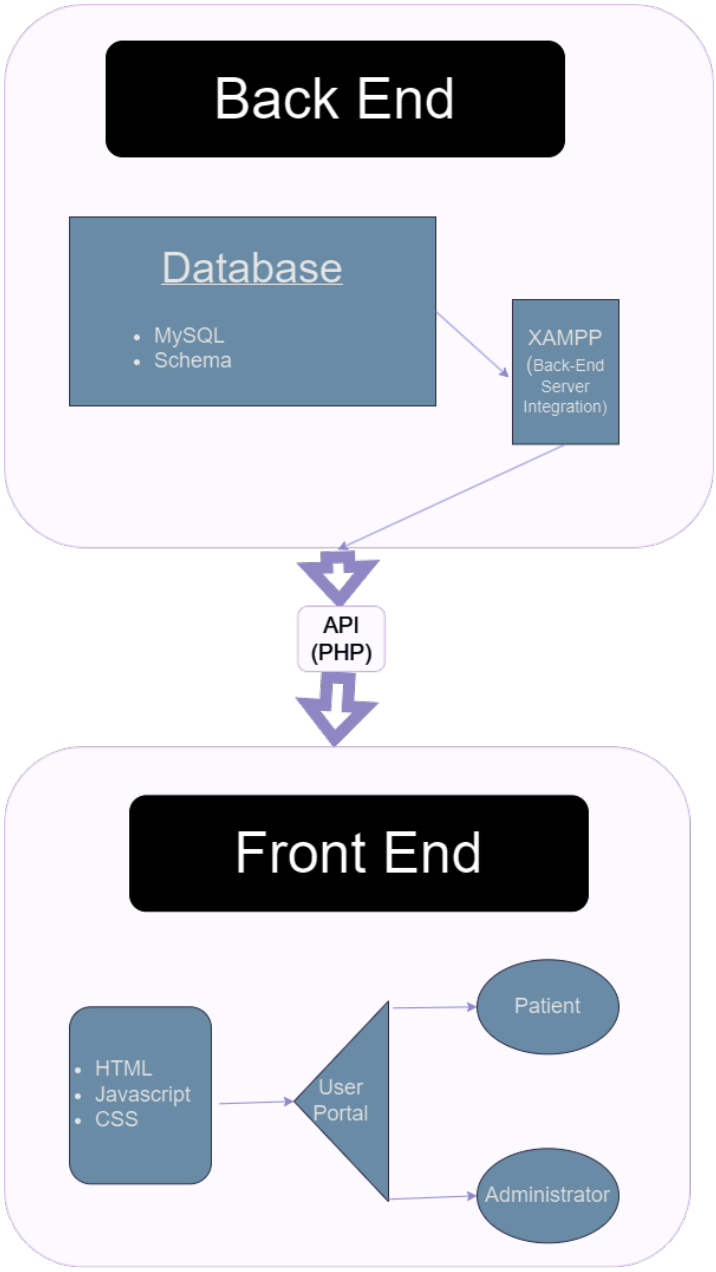

FIGURE 1. Overall structure of Project

Moreover, I think the thing that helped us the most throughout this project was the correct and concise teamwork that we all demonstrated. We all had similar understandings and work ethics when it came to this topic. A big reason was simple for our matching schedules, which lead us to all understand the workings of a database and coding for websites. Wit this understanding of the required materials we need for this project, we were able to create something we are very proud of. Given the time constraints led us to have a kind of missing front end, we still managed to spice up some of our project with some very cool features. These feature include different aspects of familiar websites like changing

username or password, adding in new information into the system, and displaying the information the user would like. All of this was constructed through the use of teamwork and understanding the material we need for this project. Given the coding required and the bonds we have created together, the HealthVault Hub project has been widely one of our favorite projects to work on. To better understand the system we have created, lets take a look at the different aspects of the design.

A. HTML

Hypertext Markup Language is a widely used language that lets a user create and define the aspects of a web page. In a nutshell, it gives the user the ability to define tables, pictures, or even text to be displayed. It is one of the most essential languages of code that we used for this project, for every page that is shown when using our website features some elements of HTML. TO better demonstrate this, we can look at the project folder and see names such as "patient_.html" and "mainpage.html". Anytime something like this happens, it is essential to understand that HTML is the backbone of that specific web page. Obviously, it helped us a lot in this project when creating the pages, yet one of the coolest features is the fact that it is so easily integrated with the likes of CSS and JavaScript. Given this, we were able to define certain elements and then go on alter and design them to however we see fit. Another cool feature of HTML, is how easy the syntax is to read and understand.

On the topic of this, let us take a look at some of the key language aspects of HTML. In an article published on the web page "mozilla", the writer states some of the head taglines needed for the code:

- <head>
- <title>
- <body>
- <header>
-
- <p>

These elements illustrate the easy and concise style that HTML adopted. With the help of this, we were able to make a front end that worked very well with our back end. We would say overall, HTML is one of the coolest languages we have ever worked with because of it's versatile and simpleness. For the project, it gave us the pages we would use for all our elements.

B. JAVASCRIPT

Moreover, JavaScript helped us give some functionality to our project. It was very useful with the main page and the login pages. JavaScript was predominately used for adding the interactive and dynamic parts of our web pages. The syntax was a bit harder to get our head wrapped around, but it is similar to the works of C++ and java. It helped us add the functionality to the buttons, the text boxes, and make things move. When we made the first parts of our project, we leaned in more heavily to JavaScript because we lacked the knowledge in PHP. However, this did not hurt us much because of its good connection with HTML. We were able to integrate it very easily. I would say overall, the majority of the functionality in the main page and login pages is through the use of this language. To further understand JavaScript, let us look at some of the syntax for it.

Furthermore, a lot of the code for JavaScript is similar to other languages, but we think one of the best features is its way of creating functions. We can create a function that directly works with HTML and lets us take information then give it back to the HTML page. This also let us directly link to PHP pages in the beginning portion for our project. Overall, we would say JavaScript is probably one of our least favorite languages we used for this project, but it was still very easy and necessary for our design. For any questions and queries of how to do things, we have a JavaScript online manual we referred to constantly in creating this project.

C. CSS

CSS or Cascading Style Sheets is usually used to design how an HTML site looks like. In our project, it was very vital for making sure everything matches colors and delivering all the information in a formal style. It is very easily integrated into to HTML, and the syntax is very fun to play with. It is also important to note that for different page in the project, there is a corresponding CSS page that designs all the information in it. To further illustrate this idea, lets take a look at our project.

In our project, we have a darker design with everything being either grey or black with white text. We did this out of personal interest, but we were able to copy the same hex value for majority of the site. For instance, we can change the back ground color for most page by adding something like this, "background-color: black;". This is the majority of the syntax for the CSS pages, so it is easy to see that integrating it into the project was very easy because of the simple syntax. We would say that overall, the use of CSS is very simple yet it is so vital to how our project looked. We have provided a reference for a basic manual to CSS below that demonstrates the usage other language further.

D. XAMPP

A requirement of the project was to display the workings of the database in some way, and allow a user to interact with it. This included running queries and viewing statistics and data that was present within the database, as well as adding new entries and editing the existing data. Consequentially, there was a need for some way to connect the database with the front end HTML we has already made.

XAMPP is a free and open-source cross platform web server solution which is developed by Apache Friends. It

includes a variety of language and software supports, but we were most interested in its SQL and PHP server hosting. XAMPP allows a developer to host a server on their local machine, then write server side code using PHP to connect to a database. XAMPP is available for installation online for free, and each group member installed it on their machine. Upon opening XAMPP, the user will be presented with a console to open and edit the configurations of multiple types of servers. For this project, we only used the SQL server and the Apache server. PHP and all front end files should be placed somewhere in the "htdocs" directory within XAMPP, as this is the root folder of the localhost server. XAMPP also offers a platform for working with databases in a streamlined and intuitive way that doesn't even require knowledge of SQL commands. Navigating to localhost within your browser gives you access to the PHPMYADMIN page, given that the SQL server was started within the XAMPP control panel. PHPMYADMIN has many quality of life features that simplify working with a database. All databases that currently exist within the directory are listed on the left side of the page. Selecting a database provides a comprehensive view of the tables that are included within it. Databases can be imported from the local machine at any time, and do not need to be created from scratch. Selecting one of these tables will display all rows and data within them.

Given that the data is largely viewable within the browser like this, PHPMYADMIN is not often recommended for large databases with tables containing thousands, or even millions of data points. However, for the purposes of this project it provided an excellent place to host our data, as our sets were relatively small. Tables can be edited and dropped from within the browser by clicking buttons. In many cases, we already had lots of SQL code written. The page contains a section for inserting SQL code instead of interacting with the browser, and this was what we used most of the time. This was especially true for insert statements. This is one of the only parts of PHPMYADMIN that we noticed to not be useful: data insertions. It was better in almost every case to write the data insertions in SQL code outside, then navigate to the SQL section and simply copy and paste it.

E. SQL

SQL stands for Structured Query Language. It is one of the most widely used languages for creating and managing relational databases. It is used for a wide variety of tasks including querying, updating, and inserting data. Additionally, it can be used for creating and removing tables. All of these features were ones that we used in the creation of the database that was used for this project. SQL is the crux of what makes the database work.

Features of SQL

- Select statements are the most commonly used type of statement in SQL. They are highly versatile, and can be used to query anything from one single cell to massive amounts of data. They can be written in the form of a string and placed within PHP code to make and return a selection of data from the database.

- Insert statements are used to add information to the table. For testing and development purposes, we front-loaded each table in the database with several lines of test data so that the select statements could be adequately tested. These statements were used to add new data as requested by the user through the web interface. One example of this can be found in the form of creating a user's login information. After the user navigates to the new login page, they are prompted with boxes to enter the new information. This information, once entered, is sent to the PHP script and a new insert query is created. The new data is then properly inserted as a new row in the corresponding table.

- Creating tables is an integral feature of SQL, and likely the most important type of statement. When a new table is created, every column within that table must be named, and constraints must be placed on them. For example, if a column in the "Personal Information" was to be a patient's social security number, some constraints must be specified when the table is created. These may be that the entered data only contains numbers and is only 9 digits in length.

F. PHP

Hypertext Preprocessor (PHP) is an HTML embedded scripting language that is used to create dynamic web pages. Due to it being supported by XAMPP and having a good reputation when it comes to database processing, we made this choice to use it to create our API for connecting our database to the website. PHP code is nested within HTML code, which allowed us to seamlessly have our front page website, which was written in HTML, communicate with the database.

The primary way in which the users enters and queries the database was done using PHP forms. These forms are written in HTML, and were used in the form of text boxes. These boxes store data and wait for a submission button to be clicked. Once information is entered and the submission is selected, that information is posted to the PHP code that is either nested within HTML, or sits in its own file. The PHP code waits until this happens, then processes the data in some way.

Within PHP, there are naively supported libraries that improve functionality with SQL. MySQLi stands for MySQL Improved and is an extension of PHP designed to provide an interface to MySQL databases, allowing developers to interact with MySQL databases using PHP. MySQLi offers various benefits, including support for prepared statements and transactions, which enhance security and performance. It provides both procedural and object-oriented interfaces for working with MySQL databases in PHP scripts. It was used

to make the connection to the database, as well as the queries.

Making queries within PHP is fairly straight forward. Once a new mysqli object is created containing the connection to the desired database, it can be passed to a query function along with a string. This string is the SQL query. It is made in the exact same format as SQL code, and uses the same keywords and processes. Once the query is made, an often large group of data is returned from the database in the form of numerous associative arrays that use a key/value reference model. This data cannot be processed at once, and must rather be viewed row by row using loops. Once the data is extracted using this method, it can easily be transferred elsewhere or displayed as a list or table. A majority of our data is displayed in HTML tables, as we felt this offered a better user experience.

IV. DISCUSSION

OVERALL the parts we created for the project gives us a brief overview of what an actual hospital website would look like. The User Interface is simple and easy for someone to use and log in. To understand this better, let us take a look at a patient portal of a major organization such as Vanderbilt. The opening User Interface to My Health At Vanderbilt is very simple, yet it features very direct key points of entry for users: a button to change the page to English, Username/Password followed by the sign in button, Use Activation Code, Create New Account, and Pay as Guest. A major corporation like this understands that everyone is different, and an important website like this needs to be very direct and straightforward. Assuming this is how most Healthcare companies operate their websites, this is how our direct approach was to building the website.

In addition to this, there is the main homepage, which should allow the user to grab their medical information and other things. This should work directly with the MySQL side of things where we have built a schema that for now covers majority of our required fields.

Limitations

In this section, we would like to discuss some of the drawbacks that were encountered throughout the process of developing this project. This was the first semester that any of our project members had used SQL or PHP. Seeing as these languages make up a very large majority of the inner workings of the website and database, learning them quickly and accurately was essential. SQL was the simpler of the two to learn. In the current state of the project, the queries that need to be performed to retrieve the required data from the database are not very complex (relatively speaking). All that was required was single layered select statements, as well as some insert and update statements. PHP was much more difficult, however. PHP forms can be very confusing for beginners just learning server side programming. On many occasions, forms would send incorrect data, or send it to the wrong file. Our intent would be to transition from one page to another, but the new page would simply execute below the old one. These kinds of issues are also magnified when working remotely with a group. Since there are several ways to send information from an HTML file to a PHP file, the whole process can be convoluted at times.

This project was the culmination of the work of three contributors over the course of one semester. Although much time and effort was put into it, only so much can be done within this short time frame with a small group such as this. Consequentially, there are a few aspects that we would have liked to have been ale to incorporate into the project to improve upon it. Due to time constraints, a majority of our efforts were focused on the functionality of the website as well as the design and relationships of the database. Consequentially, the visual aesthetic and accessibility of the website may leave a little to be desired. It is very simple, however this may be one of its strong points. We believe that over complicating a user interface with bloated icons and lots of text leads to a frustrating experience, and our website avoids that. This being said, there are some major improvements that could be made to the look of it. There are very monotone colorations and fonts, and the site does not feel very responsive. Healthvault Hub was designed to be sort of a blank slate that companies in the health field could tailor to their needs. With more time and collaboration with some companies, the look of the site could be improved dramatically.

Future Developments

One area of the project which may be noticeably lacking is security. Healthvault Hub contains two login sections: one for patients and one for staff members. These login pages use credentials that are stored within the database. This system is relatively bare bones, and once a user is successfully logged in they may change and add credentials as they please. Additionally, all private and personal information is secured behind the required entry of a social security number. While it is generally very private, having lots of information protected only using one method is not very secure. In the future, we Hope to add more security checks to protect the personal data of our users.

Anther notable security flaw that could cause a large data breach is SQL injections. A SQL injection is a type of cyberattack where an attacker exploits vulnerabilities in a website or web application to execute malicious SQL statements. These statements are typically injected through user input fields such as forms on a website. Our web interface is filled with HTML forms that directly interact with SQL through PHP. As it stands now, there are no protections in place against such kinds of attacks.

We have a couple of big ideas with regard to features that we would like to add that would enhance the user experience. The ability to schedule and check on appointments would be a very useful tool for both staff and patients. When

comparing an idea like this to the other aspects of the project, it would be relatively simple to integrate. This section would be placed in both the staff service section of the web page and the patient section. Both sections would have different designs and access to different pieces of information, but both ends would be able to create new appointments and view currently existing ones. This would create a smooth platform to generate more business and make the jobs of staff easier. Furthermore, it would allow patients to avoid the hassle of making phone calls to the business using Healthvault Hub, and rather be able to make appointments easily online.

V. CONCLUSION

ALL things considered, this project has yielded some interesting findings for us. One notable instance is the coherence between the web page and the database. The way they are created and integrated together largely depends on how well one complements the other. This setup closely resembles a basic client/server model, with a strong emphasis on the client side. These findings provide us with knowledge that will be valuable in our studies and future careers. The implications of what we have learned from this project are beneficial from all perspectives.

In turn, this project helps us understand databases applied in the healthcare world. While our version of healthcare database is something we are proud of, there is always room for improvement. We could always implement a mobile app that would directly link to the database or even develop the some sort of overall security for our web page. The possibilities our endless, but we are proud with our final results. This project served as a learning tool to understand the intricacy of the healthcare industry's components. It focused on building the database, which also incorporated a front-end portion. This portion provides a way for a user to access their information, enriching our understanding in the context of the bigger picture. Additionally, this project provided all of us with excellent practice working remotely with a group on something new. We learned a lot about communication tools, and developed new skills that will be essential when woking with others in our future careers.

VI. REFERENCES

GeeksforGeeks. (2023, April 15). CSS tutorial. GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/css-tutorial/.

MDN Contributors. (2024). HTML: HyperText Markup Language. Mozilla Developer Network. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML.

The Modern JavaScript Tutorial. (2022, April 25). The Modern JavaScript Tutorial. [Online]. Available: https://javascript.info/.

Apache Friends. (2024). Apache Friends - XAMPP. [Online]. Available: https://www.apachefriends.org/.