**Website Development with XAMPP**

Student’s name

Course name and number

Instructor’s name

Date submitted

**Website Development with XAMPP**

**Introduction**

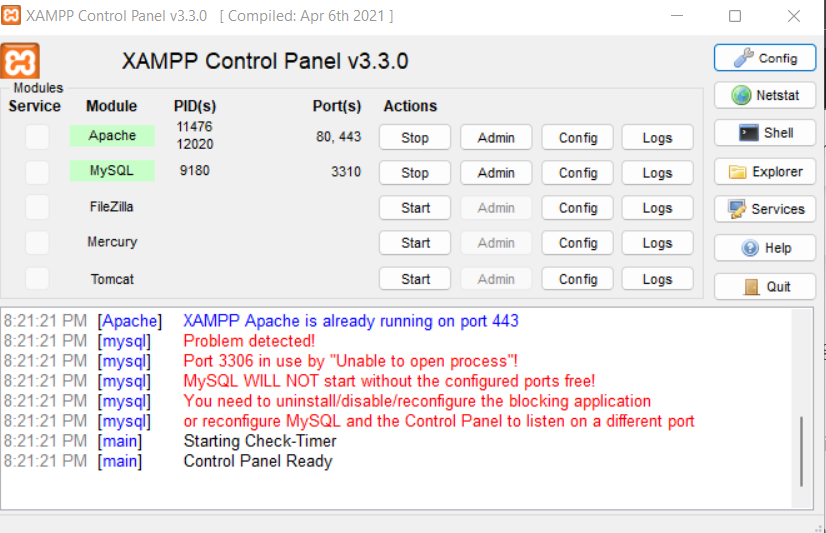
This study discusses the use of practical tools and methods utilised in website creation. We also talk about how websites are put together, with a particular emphasis on the XAMPP tool, a local host. The study takes a look at how the various components of a website are built and how they all work together to create the holistic sense of a website. For the purpose of comprehending the whole process of creating a website using the XAMPP to use.

Website development is similar to home construction in that we consult an architect regarding the blueprint, the building permit, the geological survey, and the licence from the city before beginning. The requirements for website creation must be seen in their entirety, including the designing, documentation, servers, and programming languages. The most essential component of a website is choosing a programming language. HTML and CSS are mostly used in web design. HTML proficiency is not necessary for web design. We can claim that attributes like website design, formatting, page layout strategies, graphics, multimedia, and functions for multi-page websites should be present (Pitt, 2021).

A test server should be used after programming language to view the website's layout. The cause of this is that developers use programming languages, and even if they are experts in such languages, they may still make mistakes. To see a test server's preview, server-side programming must be executed. This essay discusses the test server used with the PHP programming language and the XAMPP web development platform (Pitt, 2021).

PHP is the most widely used web backend programming language. A PHP script can be executed via the command line or as a web server module. Installing a web server like Apache and a database server like MySQL is required in order to execute PHP for the web. For running PHP apps, many web servers like WAMP & XAMPP are available. Windows supports WAMP server, and Linux and Windows also support XAMPP (Lengstorf et al., 2022). You will discover how to use the XAMPP server to run a PHP program in this study.

**How to run a PHP file in XAMPP**

After writing a program on a file and saving it in the PHP format, store the file under the ‘htdocs’ directory. Additionally, one can make any kind of folder inside the `htdocs` folder and keep our codes there. The second step is to start the servers for MySQL and Apache servers from the XAMPP control panel.

Search http://localhost in any browser to get the localhost dashboard. From the above screenshot note that Apache and MySQL are coloured green indicating that they are running.

To run the file that you created, say, ‘index.php’, on your browser on the address bar write localhost/index.php

**Student’s Database.**

A database's job is to store and arrange input data. An enterprise's various applications use databases, which are organized collections of connected data. The database keeps track of the relationships between different entities in addition to the values of their properties. A database management system (DBMS), systems software that aids in managing databases shared by numerous users, manages a database. To make it simpler to access important information, data is arranged into rows, columns, and tables and is indexed.

In this paper the database that was used was called Student database. The database is used to store data that has been scraped from a website that had details for the student’s registration. A database was used in this instance because it can store large amounts of data and the various links to the various sports activities could be indexed. The database has three tables, the links table; it stores all the links for the various the various students who have been registered in the system. Each link is unique, therefore the primary key. The second table is called the programs, it stores all the pertinent information regarding each school programs that students are enrolled into. The third table is called login, it stores all the usernames and their passwords.

**Log In, Registration and Landing Page.**

To create the Log In, Registration and Landing page, the first step was to create a connection to the MySQL database (Student dB). The connection requires one to specify the server, which in this case is the localhost, the username, password and the name of the database that one is trying to connect to.

<?php

$servername = "localhost";

$username = "root";

$password = "4156";

$db\_name = "student\_db";

$conn = new mysqli($servername, $username, $password, $db\_name);

if ($conn->connect\_error) {

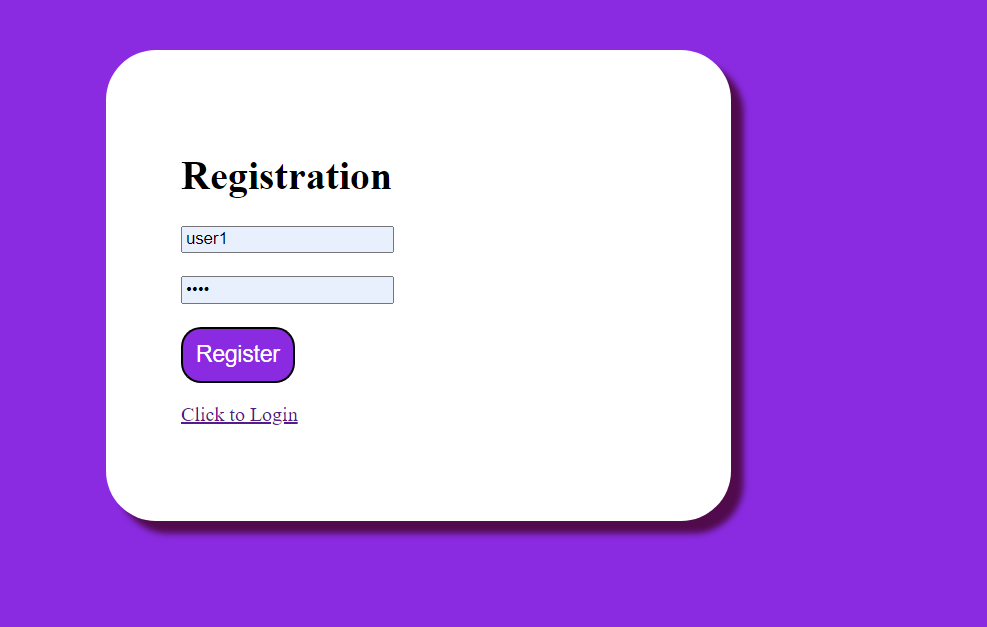
die("Connection failed".$conn->connect\_error);

}

echo "";

?>

The registration page was created by first leveraging the connection that was created earlier. The registration.php adds users to the users’ table in this case is the login table. After adding the details, one is moved to the log in page.



Below is the code for registration page:

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8"/>

    <title>Registration</title>

    <link rel="stylesheet" href="style.css"/>

</head>

<body>

<?php

    require('connection.php');

    // When form submitted, insert values into the database.

    if (isset($\_REQUEST['username'])) {

        // removes backslashes

        $username = stripslashes($\_REQUEST['username']);

        //escapes special characters in a string

        $username = mysqli\_real\_escape\_string($conn, $username);

        $password = stripslashes($\_REQUEST['password']);

        $password = mysqli\_real\_escape\_string($conn, $password);

        $query    = "INSERT into `login` (username, password)

                     VALUES ('$username', '$password')";

        $result   = mysqli\_query($conn, $query);

        if ($result) {

            echo "<div class='form'>

                  <h3>You are registered successfully.</h3><br/>

                  <p class='link'>Click here to <a href='index.php'>Login</a></p>

                  </div>";

        } else {

            echo "<div class='form'>

                  <h3>Required fields are missing.</h3><br/>

                  <p class='link'>Click here to <a href='registration.php'>registration</a> again.</p>

                  </div>";

        }

    } else {

?>

<div id = "form">

    <form class="form" action="" method="post">

        <h1 class="login-title">Registration</h1>

        <input type="text" class="login-input" name="username" placeholder="Username" required /></br></br>

        <input type="password" class="login-input" name="password" placeholder="Password"></br></br>

        <input type="submit" name="submit" value="Register" class="login-button" id = "btn">

        <p class="link"><a href="index.php">Click to Login</a></p>

    </form>

</div>

<?php

    }

?>

</body>

</html>

Below is the code for the login page:

<?php

include("connection.php");

if(isset($\_POST['submit'])){

    $username = $\_POST['user'];

    $password = $\_POST['pass'];

    $sql = "select \* from login where username = '$username' and password = '$password'";

    $result = mysqli\_query($conn, $sql);

    $row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);

    $count = mysqli\_num\_rows($result);

    if($count==1){

        header("Location:welcome.php");

    }

    else{

        echo '<script>

            window.location.href  = "index.php"

            alert("Login Failed. Invalid Username or Password")

        </script>';

    }

}

?>

The log in page verifies that the user is in the users’ table. If the user is in the login table(users table), the user is then moved to the landing page. The welcome page(landing page) just verifies that the user succeeded in log in.



Below is the code for the landing page:

<?php

    include("connection.php")

?>

<!DOCTYPE html>

<html lang="en">

    <header>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE-edge">

        <meta name="viewport" content="width=device-width", initial-scale =1.0>

        <title>Login Page</title>

        <link rel="stylesheet" type ="text/css" href="style.css">

    </header>

    <body>

        <div id = "form">

            <h1>Login</h1>

            <form name="form" action = "login.php" onsubmit="return isvalid()" method="post">

                <label>username: </label>

                <input type="text" id="user" name="user"/></br></br>

                <label>Password :</label>

                <input type="password" id="pass" name="pass"/></br></br>

                <input type="submit" id="btn" value="Login" name="submit"/>

            </form>

            <p class="link"><a href="registration.php">Not a user, Register.</a></p>

        </div>

        <script>

            function isvalid(){

                var user = document.form.user.value;

                var pass = document.form.pass.value;

                if (user.length=="" && pass.length==""){

                    alert("Username and Password Fields are Empty");

                    return false;

                }

                else{

                    if (user.length==""){

                    alert("Username is Empty");

                    return false;

                }

                if (pass.length==""){

                    alert("Password is Empty");

                    return false;

                }

                }

            }

        </script>

    </body>

</html>

**Conclusion**

Website development is all about connecting the backend(the database) and the front end; how to information in the database is displayed to and accessed by the various users. XAMPP is the easiest way to install and use PHP and MYSQL. PHP is the most popular programming language for website development.

**References**

Pitt, C. (2021). Ways to Use PHP. In *Pro PHP 8 MVC* (pp. 1-12). Apress, Berkeley, CA.

Lengstorf, J., Blom Hansen, T., & Prettyman, S. (2022). Getting Ready to Program. In *PHP 8 for Absolute Beginners* (pp. 3-37). Apress, Berkeley, CA.