

TicTacToe - Web Application in PHP

BY LIZ ORIANA RODRIGUES CRUZ

After taking the course "Building Web Applications in PHP" given by Dr. Charles Russell Severance I was motivated to do the next project that I will explain you through a series of steps that will lead to the final result in a successful way.

Installations

First of all you must choose a local web server that will allow you to run the files, There are several programs depending on the operating system you have, MAMP for Mac OS or Windows and XAMPP for Windows devices. Below are the links for installation.

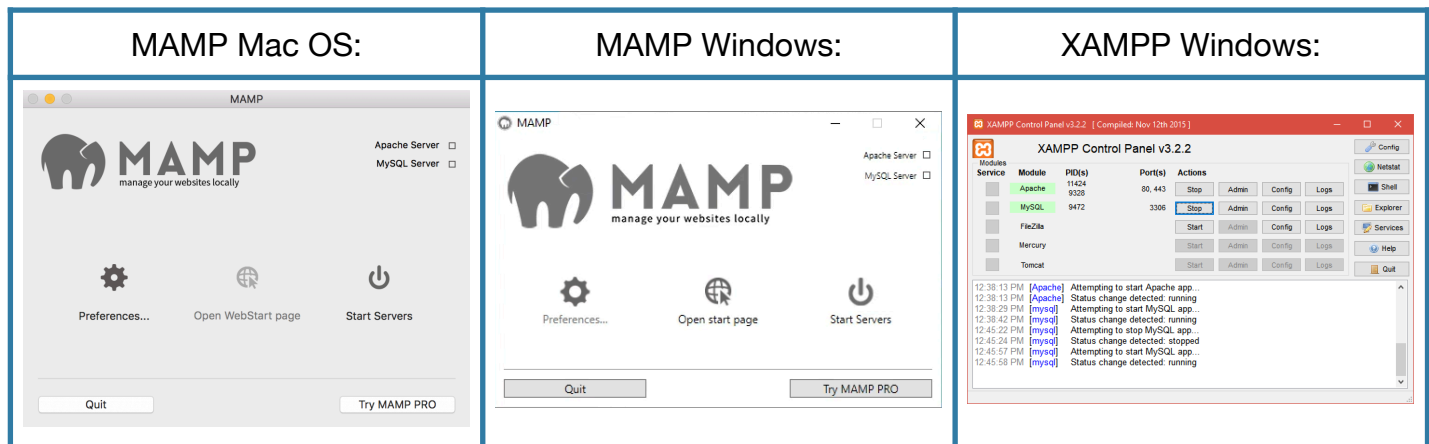
MAMP Mac OS: <https://www.mamp.info/de/mac/>

MAMP Windows: <https://www.mamp.info/en/windows/>

XAMPP Windows: <https://www.apachefriends.org/index.html>

Use of the programs

When the installation is done properly, you should start running the program and it will look like this:



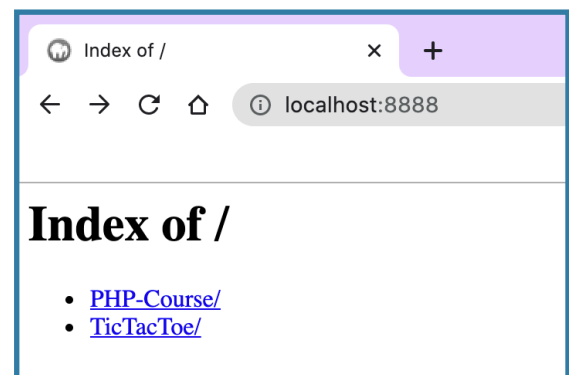
Now, how do you start viewing your files? It's very easy, first your files must be stored in the location:

MAMP Mac OS: `/Applications/MAMP/htdocs`

MAMP Windows: `C:\MAMP\htdocs`

XAMPP Windows: `C:\XAMP\htdocs`

Then in your browser you will enter `http://localhost:8888/` which at first glance will show you your folders and files located at the address mentioned above for each program and operating system.



Previous Knowledge

The following definitions will help you understand many of the web programming, design and domain languages that will be worked on throughout the document, although it is also necessary that you have at least a basic knowledge of each of them to develop the tutorial.

○ HTML (*HyperText Markup Language*):

Is a markup language created in 1991 that is used to define the meaning and structure of the content of a web page that is composed through tags of which these are the main ones along with its structure:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Metadata about the page -->
    <!-- This is usually the title of the page,
         the tag that adds the CSS document to the file, etc. -->
  </head>
  <body>
    <!-- Displayable content of the page -->
    <!-- Here is usually the page content,
         forms, lists, tables, paragraphs, etc. -->
  </body>
</html>
```

○ CSS (*Cascading Style Sheets*):

Is a design language or styles proposed in 1994 but its first presentation was in 1997, this language defines how the aesthetic elements of the page should be shown as position, shape, spacing, colors, etc. This language is used for HTML and XML type documents.

```
body{
  /* Customize everything in the body tag
    of the HTML or XML document */
}

p{
  /* Customize everything in the paragraph tag
    of the HTML or XML document */
}

.Change{
  /* Customize any label that uses the class attribute
    with this name of the HTML or XML document */
}
```

○ PHP (*Hypertext Preprocessor*):

Is the programming language introduced in 1994 for web development which originally meant "Personal Home Page", this language extends HTML pages by adding segments of code.

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <?php
      // Code implemented in PHP within a world of HTML tags
      // This is a .php file
    ?>
  </body>
</html>
```

○ SQL (*Structured Query Language*):

Is an interactive standard programming language that helps to solve specific problems, manipulation and integrity of information and to obtain information from a database.

○ Apache HTTP Server:

Is an open source HTTP web server that was developed in 1995 for the creation of web pages and services. It is a multi-platform server, free and noted for its security and performance.

Implementation of Login

You start with the file named `index.php`, which was a login for the single game which will receive any username but will have a password for everyone.

○ Procedure in PHP.

1. Create variables for each characteristic:
 - a. Series of random letters, numbers and symbols you want.
 - b. The password you wish to use to access the game.
 - c. Using the cryptographic reduction function `md5` to apply a concatenation of the above variables.
 - d. It starts false as it will change if the password fails.
2. Make an `if` to verify if the `POST` data is in place:
 - a. Create a variable to use the `hash` function with the `md5` form and the concatenation of the first variable created and the `POST` pass data.
 - b. Make another `if` to check if the two variables with functions are the same
(So the program will know if the inserted password is the correct one)
 - c. Do an `else` in case the password is not correct to send a message to the user.

```
$salt = 'dju@si_1';  
$pass = 'tictactoe123';  
$pass_hash = md5($salt.$pass);  
$fail = false;
```

```
if ( isset($_POST['user']) && isset($_POST['pass']) ) {  
    $check = hash('md5', $salt . $_POST['pass']);  
  
    if ( $check == $pass_hash ) {  
        header("Location: game.php?user=".urlencode($_POST['user']));  
        return;  
    } else {  
        $fail = "Incorrect password";  
    }  
}
```

○ Procedure in HTML.

1. Write the main HTML tags.
2. Inside the `<head>` tag you must:
 - a. The title of the page.
 - b. The image you want to have on the thumbnail.
 - c. Permission to use the `.css` file in the document.
3. Inside the `<body>` tag you must:
 - a. Enter the image of the game title.
 - b. Create the form with the `POST` method that it should be:
 - I. With a `<p>` tag type Username and password.
 - II. With a `<input>` tag create the space to write the user and password.
 - c. Create the button that will lead to the game.
 - d. Use the `` tag to change the "Incorrect password" message to HTML.

```
<head>  
    <meta charset="UTF-8">  
    <title> iTicTacToe! </title>  
    <link rel="shortcut icon" href="img/MiniLogo.png" />  
    <link rel="stylesheet" href="style.css" />  
</head>
```

```
<body>  
      
    <div class="box-login">  
        <form method="POST">  
            <label class="text-login">  
                <input class="input-user" id="user" name="user" type="text" required>  
            </label>  
            <label class="text-login">  
                <input class="input-pass" id="pass" name="pass" type="password" required>  
            </label>  
            <input class="button-login" type="submit" value="Log in"/>  
            <span class="pass-incorrect"> <?htmlentities($fail)?> </span>  
        </form>  
    </div>  
</body>
```

Implementation of Game

Now the explanation goes to the file named **game.php** which will welcome the player, show the game, tell you what your position was at the end of the game against the computer and let you log out when you want to finish.

o Procedure in PHP.

1. Create variables for each characteristic:

- The information on how the game ends and as long as it doesn't end will be nobody "n"
- An arrangement with the game spaces, 9 spaces.

```
$winner = 'n';  
$box = array ('',' ',' ',' ',' ',' ',' ',' ',' ');
```

2. Make an **if** that will see the post data to give name to each space of the arrangement.

(All of the following go into this **if**)

```
if (isset($_POST['submitbtn'])) {  
  
    $box[0] = $_POST["box0"];  
    $box[1] = $_POST["box1"];  
    $box[2] = $_POST["box2"];  
    $box[3] = $_POST["box3"];  
    $box[4] = $_POST["box4"];  
    $box[5] = $_POST["box5"];  
    $box[6] = $_POST["box6"];  
    $box[7] = $_POST["box7"];  
    $box[8] = $_POST["box8"];
```

3. Make an **if** to check if three "x's" are placed for the player win.

- The first variable created will have the value **x**
- The image will be displayed that will tell the player that he won.

```
if (($box[0] == 'x' && $box[1] == 'x' && $box[2] == 'x') ||  
    ($box[3] == 'x' && $box[4] == 'x' && $box[5] == 'x') ||  
    ($box[6] == 'x' && $box[7] == 'x' && $box[8] == 'x') ||  
    ($box[0] == 'x' && $box[3] == 'x' && $box[6] == 'x') ||  
    ($box[1] == 'x' && $box[4] == 'x' && $box[7] == 'x') ||  
    ($box[2] == 'x' && $box[5] == 'x' && $box[8] == 'x') ||  
    ($box[0] == 'x' && $box[4] == 'x' && $box[8] == 'x') ||  
    ($box[2] == 'x' && $box[4] == 'x' && $box[6] == 'x')) {  
    $winner = 'x';  
    print("<br><br><img class = 'Results' src = 'img/Win.png' alt = 'Win'> </br>");  
}
```

4. Make a **for** to give the computer permission to play.

```
$temp = 0;  
for ($i = 0; $i <= 8; $i++){  
    if ($box[$i] == ''){  
        $temp = 1;  
    }  
}
```

5. Make an **if** that checks whether the game is not finished and whether the computer is allowed to play and in develop:

- Create several **if**'s that will help the computer with its turn when the player is one move away from winning.
 - If** the player is not close to winning the computer will play in a random position for a **while**.
- (All of the following go into this **if**)

```
if ($temp == 1 && $winner == 'n') {  
    $i = rand() % 8;  
  
    if (($box[3] == 'x' && $box[6] == 'x') || ($box[4] == 'x' && $box[8] == 'x') || ($box[1] == 'x' && $box[2] == 'x')) {  
        $box[0] = 'o';  
    } else if (($box[0] == 'x' && $box[2] == 'x') || ($box[4] == 'x' && $box[7] == 'x')) {  
        $box[1] = 'o';  
    } else if (($box[0] == 'x' && $box[1] == 'x') || ($box[5] == 'x' && $box[8] == 'x') || ($box[4] == 'x' && $box[6] == 'x')) {  
        $box[2] = 'o';  
    } else if (($box[0] == 'x' && $box[6] == 'x') || ($box[4] == 'x' && $box[5] == 'x')) {  
        $box[3] = 'o';  
    } else if (($box[0] == 'x' && $box[8] == 'x') || ($box[2] == 'x' && $box[6] == 'x') || ($box[1] == 'x' && $box[7] == 'x') || ($box[3] == 'x' && $box[5] == 'x')) {  
        $box[4] = 'o';  
    } else if (($box[2] == 'x' && $box[8] == 'x') || ($box[3] == 'x' && $box[4] == 'x')) {  
        $box[5] = 'o';  
    } else if (($box[0] == 'x' && $box[3] == 'x') || ($box[2] == 'x' && $box[4] == 'x') || ($box[7] == 'x' && $box[8] == 'x')) {  
        $box[6] = 'o';  
    } else if (($box[6] == 'x' && $box[8] == 'x') || ($box[1] == 'x' && $box[4] == 'x')) {  
        $box[7] = 'o';  
    } else if (($box[0] == 'x' && $box[4] == 'x') || ($box[2] == 'x' && $box[5] == 'x') || ($box[6] == 'x' && $box[7] == 'x')) {  
        $box[8] = 'o';  
    }  
}
```

```
else {  
  
    while ($box[$i] != '') {  
        $i = rand() % 8;  
    }  
    $box[$i] = 'o';  
}
```

6. Create an **if** to check if three "o's" are placed for computer wins
 - a. The first variable created will have the value o .
 - b. The image will be displayed that will tell the player that he lose.

```
if (($box[0] == 'o' && $box[1] == 'o' && $box[2] == 'o') ||
    ($box[3] == 'o' && $box[4] == 'o' && $box[5] == 'o') ||
    ($box[6] == 'o' && $box[7] == 'o' && $box[8] == 'o') ||
    ($box[0] == 'o' && $box[3] == 'o' && $box[6] == 'o') ||
    ($box[1] == 'o' && $box[4] == 'o' && $box[7] == 'o') ||
    ($box[2] == 'o' && $box[5] == 'o' && $box[8] == 'o') ||
    ($box[0] == 'o' && $box[4] == 'o' && $box[8] == 'o') ||
    ($box[2] == 'o' && $box[4] == 'o' && $box[6] == 'o')){
    $winner = 'o';
    print("</br></br> <img class = 'Results' src = 'img/Lose.png' alt = 'Lose'> ");
}
```

7. Create an **else if** that checks if nobody won to send the tie image.

```
else if ($winner == 'n'){
    $winner = 't';
    print("</br> <img class = 'Results' src = 'img/Tie.png' alt = 'Tie'>");
}
```

o Procedure in HTML.

1. Write the main HTML tags.
2. Inside the **<head>** tag you must:
 - a. The title of the page.
 - b. The image you want to have on the thumbnail.
 - c. Permission to use the .css file in the document

```
<head>
  <meta charset="UTF-8">
  <title> iTicTacToe! - Game</title>

  <link rel="shortcut icon" href="img/MiniLogo.png" />

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

3. Inside the **<body>** tag you must:
 - a. Create labels for top decoration.
 - I. Give a welcome message.
 - II. Create the logout button.

```
<body>
  <div class = "Upper-Bar-W">
    <div class = "Upper-Bar-B">
      <div class = "Upper-Bar-P">
        <h1> iWelcome! </h1>
        <a class = "button-logout" href="index.php">Log out</a>
      </div>
    </div>
  </div>
```

- b. Make a form with the post method for the game box:
 - I. Do a small part in PHP
 - Create a **for** for displaying the computer's moves.
 - Create a button to allow the computer to move if it has not finished the game.
 - Create a button for yourself and the game will start again.

```
<form name="tictactoe" action = "game.php" method = "post">
  <?php
    for ($i = 0; $i <= 8; $i++){
      printf('<input type = "text" name= "box%s" value = "%s" class = "box-game">', $i, $box[$i] );
      if ($i == 2 || $i == 5 || $i == 8){
        print('<br>');
      }
    }

    if ($winner == 'n'){
      print('<br> <input type = "submit" name = "submitbtn" value = "Next" class = "button-next">');
    } else {
      print('<br><input type = "button" name = "newgamebtn" value = "Play Again"
        class = "button-again" onclick = "window.location.href=\''game.php\'' ">');
    }
  ?>
</form>
</body>
```

Implementation in CSS

Finally I will show the document `style.css` which is in charge of customizing all the documents shown above and in which document they were used.

- For all documents.

```
body{
    background-color: #c2c7fd;
    font-family: Helvetica, Arial, sans-serif;
    line-height: 1.5;
}
h1{
    font-size: 50px;
    font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
    position: absolute;
    top: 0px;
    left: 700px;
}
```

- For the `index.php` file.

```
/* ----- THIS IS DECORATION FOR THE BUTTON ----- */
.button-login {
    color: white !important;
    font-size: 20px;
    font-weight: 500;
    padding: 0.5em 1.2em;
    background: #ffc71a6;
    border: 4px solid #853553;
    border-radius: 6px;
    transition: all 1s ease;
    position: relative;
}
.button-login:hover {
    background: #4cc4ff;
    color: rgb(0, 0, 0) !important;
    border: 4px solid #175c7e;
}
/* ----- THIS IS DECORATION FOR THE FORMS BARS ----- */
::-ms-input-user,
input[type='text']:active:focus,
input[type='text']:hover{
    outline : none;
    font-size:20px;
}
::-ms-input-pass,
input[type='password']:active:focus,
input[type='password']:hover{
    animation: backwards;
    outline : none;
    font-size:20px;
}
```

```
/*----- THIS IS DECORATION FOR TITTLE IN IMAGE -----*/
.title{
    float: inherit;
    width: 500px;
    position: absolute;
    top: 150px;
    left: 600px;
}
/* ----- THIS IS DECORATION FOR BOX LOG IN ----- */
.box-login{
    position: absolute;
    top: 300px;
    left: 650px;
    background-color: white;
    text-align: center;
    height: 300px;
    width: 300px;
    padding: 40px;
    border-top-left-radius: 50%;
    border-top-right-radius: 50%;
    border-bottom-right-radius: 50%;
    border-bottom-left-radius: 10%;
}
/* ----- */
/*----- THIS IS DECORATION FOR THE TEXTS IN THE BOX -----*/
.text-login{
    font-size:25px;
    font-family: Impact;
    font-weight: 1200;
    line-height: 0.8;
}
.pass-incorrect{
    font-size: 15px;
    font-family: 'Gill Sans';
}
```

- For the `game.php` file.

```
/*----- THIS IS DECORATION IN THE PAGE -----*/
.Upper-Bar-W{
    position: absolute;
    top: 0px;
    left: 0px;
    background-color: white;
    height: 50px;
    width: 1598px;
    padding: 40px;
    box-shadow: 4px 4px 10px #666;
}
.Upper-Bar-B{
    position: absolute;
    top: 0px;
    left: 0px;
    background-color: #4cc4ff;
    height: 20px;
    width: 1619px;
    padding: 30px;
    box-shadow: 4px 4px 10px #666;
}
.Upper-Bar-P{
    position: absolute;
    top: 0px;
    left: 0px;
    background-color: #fc71a6;
    height: 0px;
    width: 1639px;
    padding: 20px;
    box-shadow: 4px 4px 10px #666;
}
```

```
/* ----- THIS IS DECORATION FOR BOXS GAME ----- */
.box-game{
    background-color: white;
    border: 3px solid #6f73a7;
    width: 150px;
    height: 150px;
    font-size: 70px;
    text-align: center;
    position: relative;
    top: 200px;
    left: 800px;
}
/* ----- */
/*----- THIS IS DECORATION FOR RESULTS IN THE GAME -----*/
.Results{
    width: 600px;
    float: inherit;
    position: absolute;
    top: 220px;
    left: 10px;
}
```

```
/* ----- THIS IS DECORATION FOR THE BUTTONS ----- */
.button-logout{
    color: white;
    text-decoration: none;
    font-size: 10px;
    font-weight: 10;
    padding: 0.5em 1.2em;
    background: black;
    border-radius: 6px;
    position: absolute;
    top: 92px;
    left: 1680px;
}
.button-next{
    color: white;
    font-size: 40px;
    font-weight: 50;
    padding: 0.5em 1.2em;
    background: #ffc71a6;
    border: 4px solid #853553;
    border-radius: 6px;
    transition: all 1s ease;
    position: absolute;
    top: 720px;
    left: 950px;
}
.button-again{
    color: white !important;
    font-size: 40px;
    font-weight: 50;
    padding: 0.5em 1.2em;
    background: #4cc4ff;
    border: 4px solid #175c7e;
    border-radius: 6px;
    transition: all 1s ease;
    position: absolute;
    top: 720px;
    left: 152px;
}
```