XML and PHP

A description of the site.

The site that I have decided to develop for this ePortfolio is a site that displays a list of Puns, or Jokes to the user. The user is able to view these puns and decide which one they find to be the best, evidently, allowing them to vote for their favorite pun! After voting for the pun the user likes the best, the site will save that information and update the data to reflect that another user has voted for that specific pun. Then, the site will direct the user to a second page, where they can view the Pun's standings, and see which ones have the most/least votes, according to how other users have voted.

The web technologies from above that you used and why.

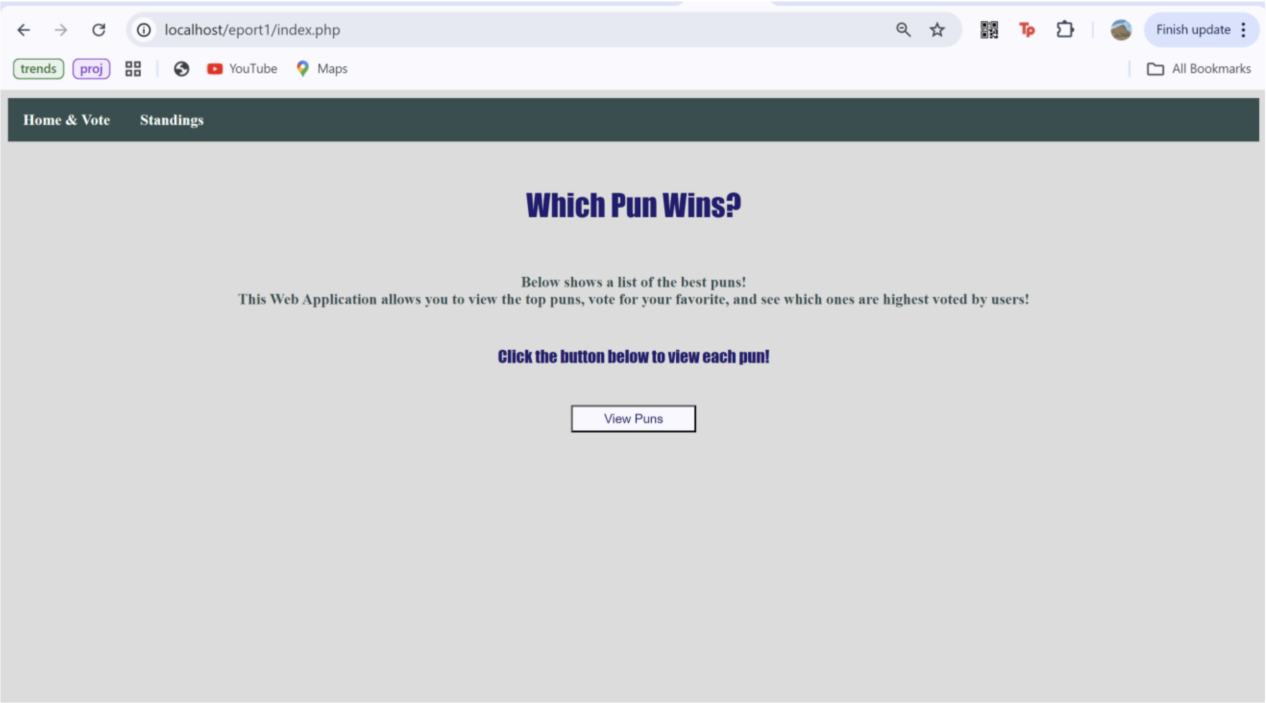
I have decided to make this website with a compilation of various technologies. The site runs on a locally hosted PHP server; therefore, the two main webpages are PHP webpages, with HTML embedded. I chose the PHP web environment because it is a familiar programming language for me, however, I had never integrated XML or JSON along with PHP before. I wanted to challenge myself, so that I can continue learning, and expand my knowledge within this programming language, hence, I decided to use PHP. Additionally, the data interchange format I decided to use is XML. I decided on XML due to its well-structured format, and clear organization. Also, I figured that since we had been focusing on JSON so much in the previous labs, I wanted to use XML as an additional challenge, to really grow in my understanding. I used AJAX in my JavaScript to demonstrate an Asynchronous retrieval and display of my XML data, to showcase my recently acquired understanding of this technique from the recent module.

Finally, I integrated the powerful jQuery library within my JavaScript to facilitate development, and make the source code neater, and more visually pleasing.

Screenshots of your website.

The following screenshots demonstrate the output of my website, with the respective file names, as well as a brief description of how each element works, for a clearer understanding.

Landing Page (index.php):

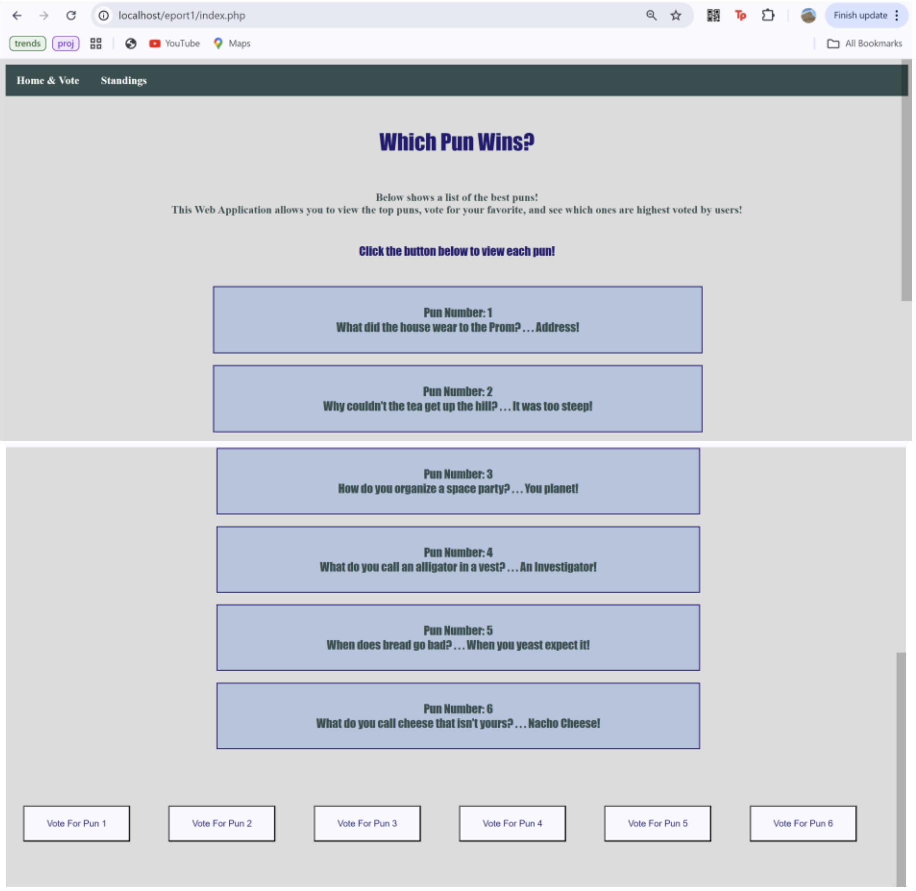


Which Pun Wins?

Below shows a list of the best puns!
This Web Application allows you to view the top puns, vote for your favorite, and see which ones are highest voted by users!

Click the button below to view each pun!

Upon Clicking "View Puns" button, the xml document data is retrieved and displayed Asynchronously, as demonstrated below:



Which Pun Wins?

Below shows a list of the best puns!
This Web Application allows you to view the top puns, vote for your favorite, and see which ones are highest voted by users!

Click the button below to view each pun!

Pun Number: 1
What did the house wear to the Prom? . . . Address!

Pun Number: 2
Why couldn't the tea get up the hill? . . . It was too steep!

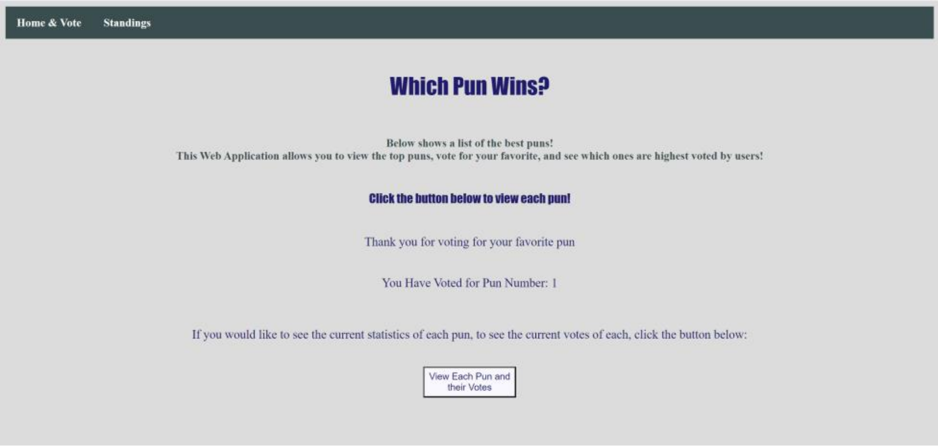
Pun Number: 3
How do you organize a space party? . . . You planet!

Pun Number: 4
What do you call an alligator in a vest? . . . An Investigator!

Pun Number: 5
When does bread go bad? . . . When you yeast expect it!

Pun Number: 6
What do you call cheese that isn't yours? . . . Nacho Cheese!

Upon clicking the button for which pun the user would like to vote for – their vote is stored in a PHP session variable, then a vote is incremented and reflected in the XML for whichever pun they chose to vote for. For example, here I chose to vote for pun number 1:



Which Pun Wins?

Below shows a list of the best puns!
This Web Application allows you to view the top puns, vote for your favorite, and see which ones are highest voted by users!

Click the button below to view each pun!

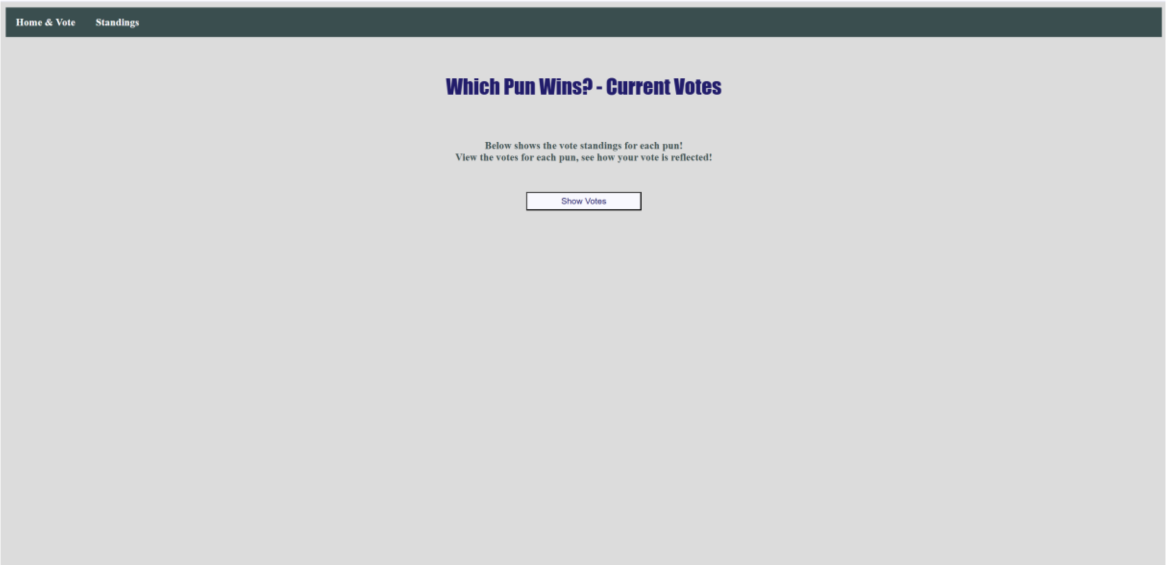
Thank you for voting for your favorite pun

You Have Voted for Pun Number: 1

If you would like to see the current statistics of each pun, to see the current votes of each, click the button below:

Upon clicking the “View Each Pun and Their Votes Button”, the user is redirected to the votes.php page, as shown in the screenshot below:

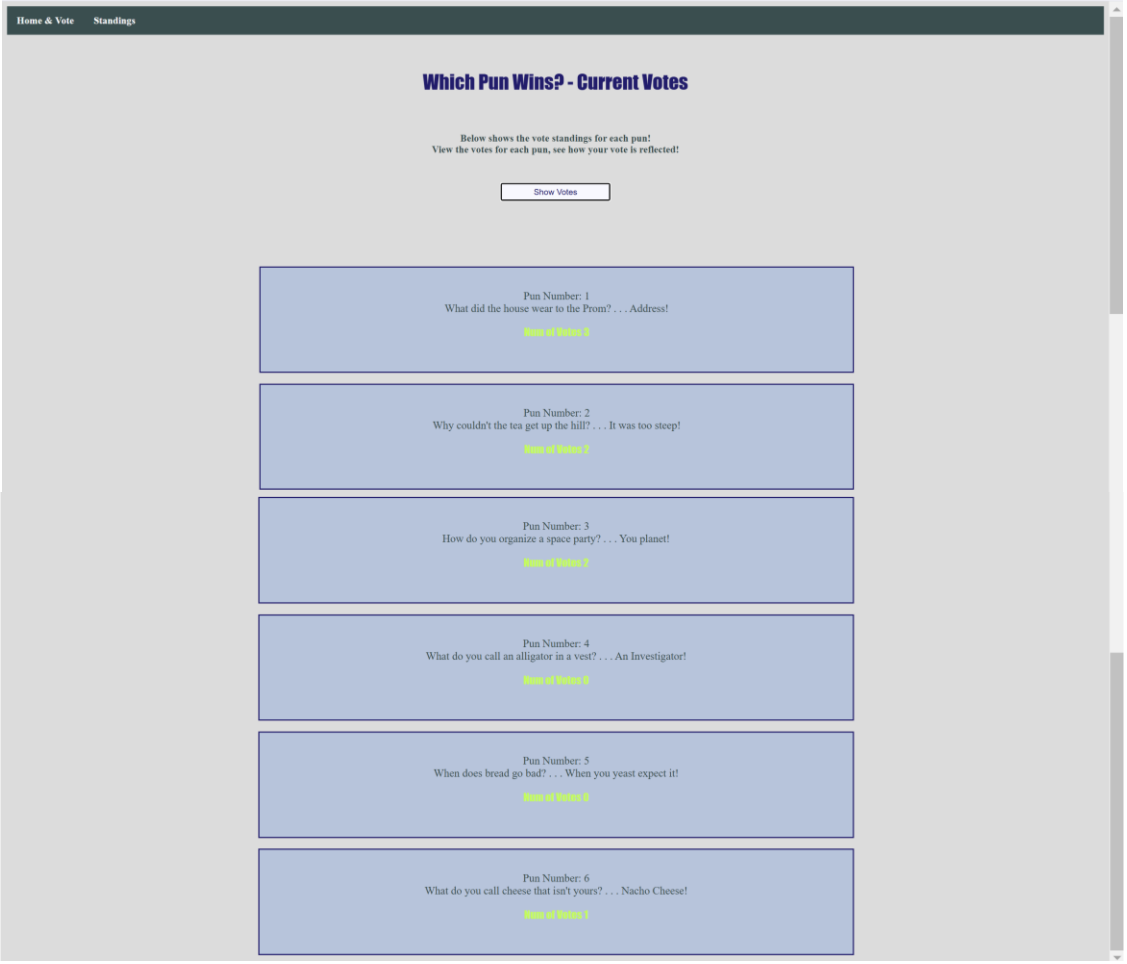
(votes.php)



Which Pun Wins? - Current Votes

Below shows the vote standings for each pun!
View the votes for each pun, see how your vote is reflected!

Upon clicking “Show Votes” button, the XML data along with the newly updated votes are retrieved and displayed, through the use of an AJAX request. The AJAX collects the XML data from a file called get.php, which gets the current data from the XML file. This then displays the following page, reflecting the correct number of votes. The result is shown below:



Which Pun Wins? - Current Votes

Below shows the vote standings for each pun!
View the votes for each pun, see how your vote is reflected!

Pun Number: 1
What did the house wear to the Prom? . . . Address!
Num of Votes 3

Pun Number: 2
Why couldn't the tea get up the hill? . . . It was too steep!
Num of Votes 2

Pun Number: 3
How do you organize a space party? . . . You planet!
Num of Votes 2

Pun Number: 4
What do you call an alligator in a vest? . . . An Investigator!
Num of Votes 0

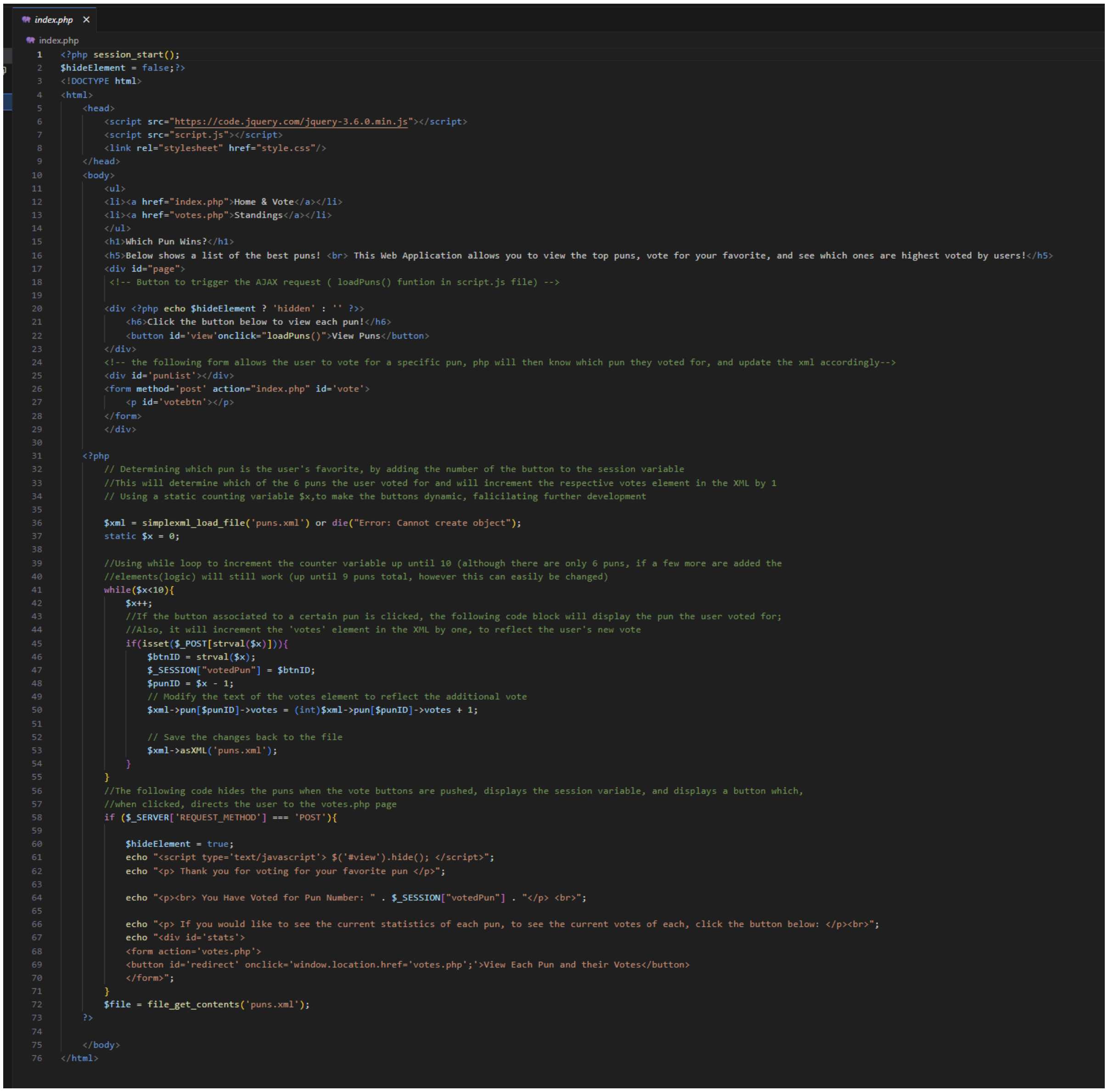
Pun Number: 5
When does bread go bad? . . . When you yeast expect it!
Num of Votes 0

Pun Number: 6
What do you call cheese that isn't yours? . . . Nacho Cheese!
Num of Votes 1

Source Code:

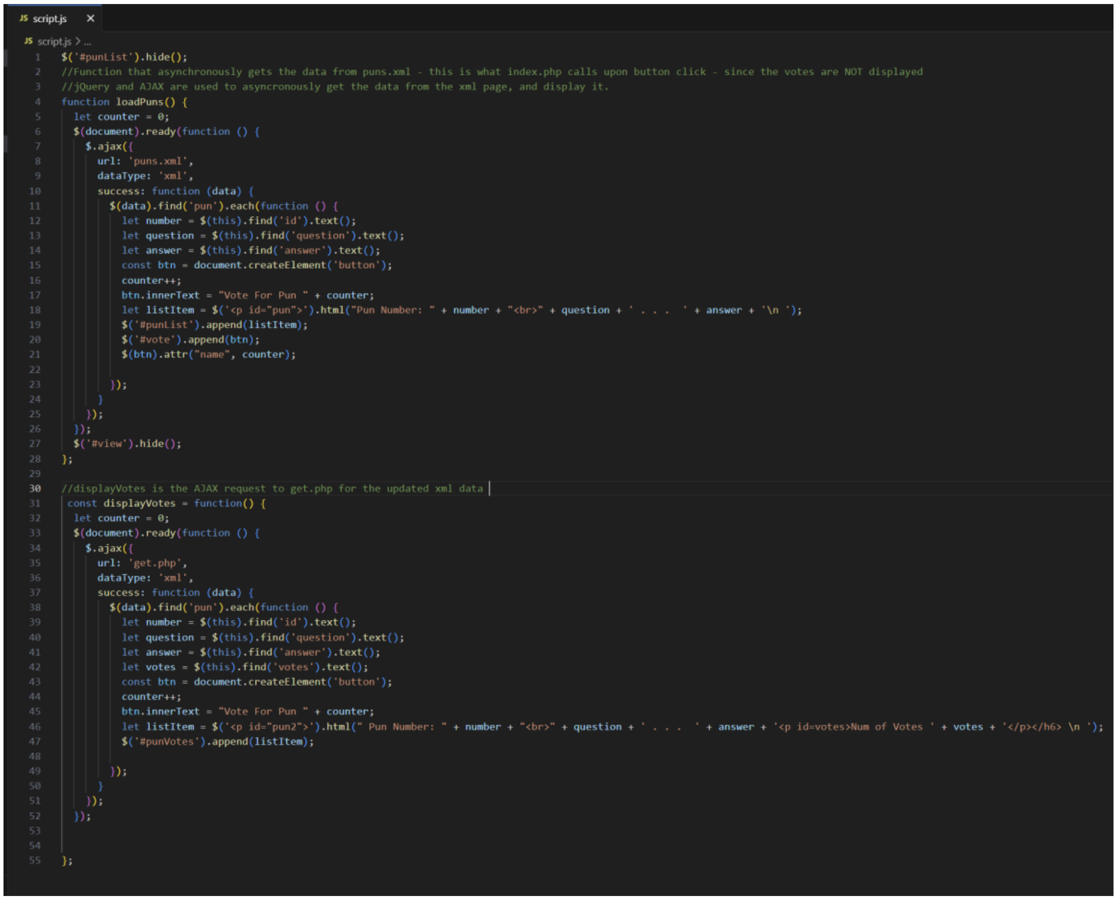
The following screenshots demonstrate the source code of each of the files developed to accomplish this website.

Index.php:



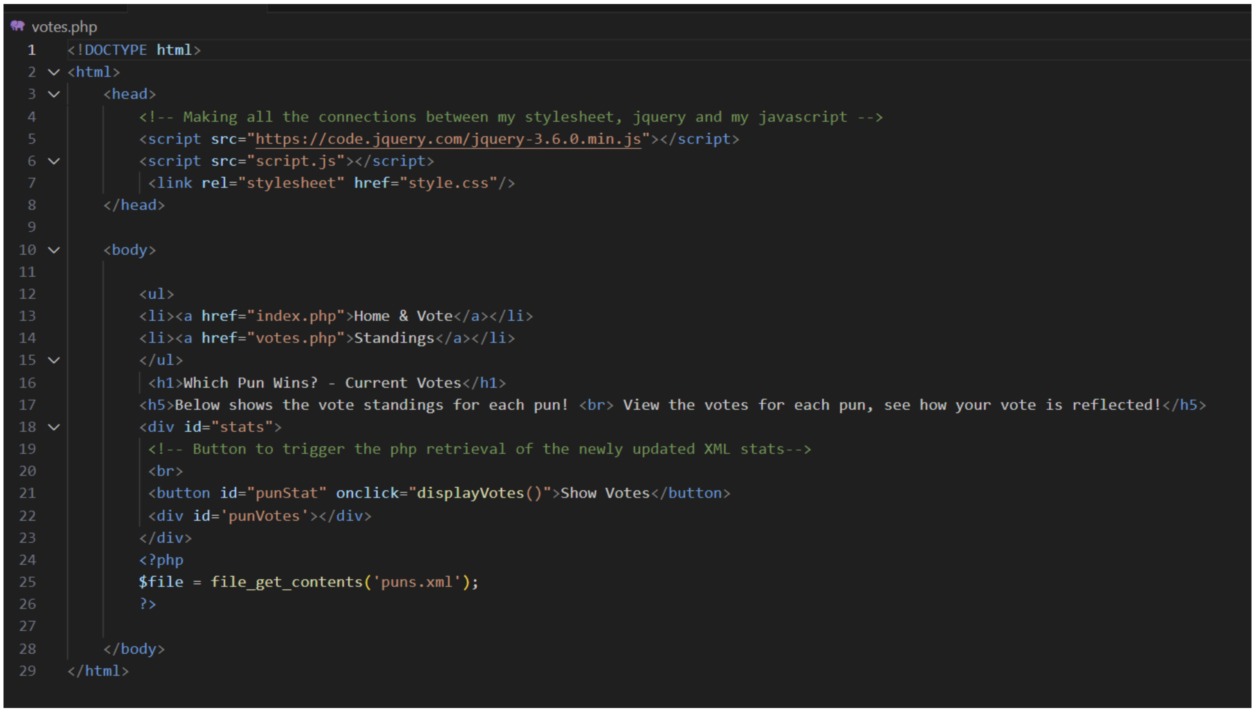
```
<?php session_start();
$hideElement = false;?>
<!DOCTYPE html>
<html>
    <head>
        <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
        <script src="script.js"></script>
        <link rel="stylesheet" href="style.css"/>
    </head>
    <body>
        <ul>
        <li><a href="index.php">Home & Vote</a></li>
        <li><a href="votes.php">Standings</a></li>
        </ul>
        <h1>Which Pun Wins?</h1>
        <h5>Below shows a list of the best puns! <br> This Web Application allows you to view the top puns, vote for your favorite, and see which ones are highest voted by users!</h5>
        <div id="page">
         <!-- Button to trigger the AJAX request ( loadPuns() funtion in script.js file) -->

        <div <?php echo $hideElement ? 'hidden' : '' ?>>
            <h6>Click the button below to view each pun!</h6>
            <button id='view'onclick="loadPuns()">View Puns</button>
        </div>
        <!-- the following form allows the user to vote for a specific pun, php will then know which pun they voted for, and update the xml accordingly-->
        <div id='punList'></div>
        <form method='post' action="index.php" id='vote'>
            <p id='votebtn'></p>
        </form>
        </div>

    <?php
        // Determining which pun is the user's favorite, by adding the number of the button to the session variable
        //This will determine which of the 6 puns the user voted for and will increment the respective votes element in the XML by 1
        // Using a static counting variable $x,to make the buttons dynamic, falicilating further development

        $xml = simplexml_load_file('puns.xml') or die("Error: Cannot create object");
        static $x = 0;

        //Using while loop to increment the counter variable up until 10 (although there are only 6 puns, if a few more are added the
        //elements(logic) will still work (up until 9 puns total, however this can easily be changed)
        while($x<10){
            $x++;
            //If the button associated to a certain pun is clicked, the following code block will display the pun the user voted for;
            //Also, it will increment the 'votes' element in the XML by one, to reflect the user's new vote
            if(isset($_POST[strval($x)])){
                $btnID = strval($x);
                $_SESSION["votedPun"] = $btnID;
                $punID = $x - 1;
                // Modify the text of the votes element to reflect the additional vote
                $xml->pun[$punID]->votes = (int)$xml->pun[$punID]->votes + 1;

                // Save the changes back to the file
                $xml->asXML('puns.xml');
            }
        }
        //The following code hides the puns when the vote buttons are pushed, displays the session variable, and displays a button which,
        //when clicked, directs the user to the votes.php page
        if ($_SERVER['REQUEST_METHOD'] === 'POST'){

            $hideElement = true;
            echo "<script type='text/javascript'> $('#view').hide(); </script>";
            echo "<p> Thank you for voting for your favorite pun </p>";

            echo "<p><br> You Have Voted for Pun Number: " . $_SESSION["votedPun"] . "</p> <br>";

            echo "<p> If you would like to see the current statistics of each pun, to see the current votes of each, click the button below: </p><br>";
            echo "<div id='stats'>
            <form action='votes.php'>
            <button id='redirect' onclick='window.location.href='votes.php';'>View Each Pun and their Votes</button>
            </form>";
        }
        $file = file_get_contents('puns.xml');
    ?>

    </body>
</html>
```

Script.js (Asynchronous requests made in the functions here)



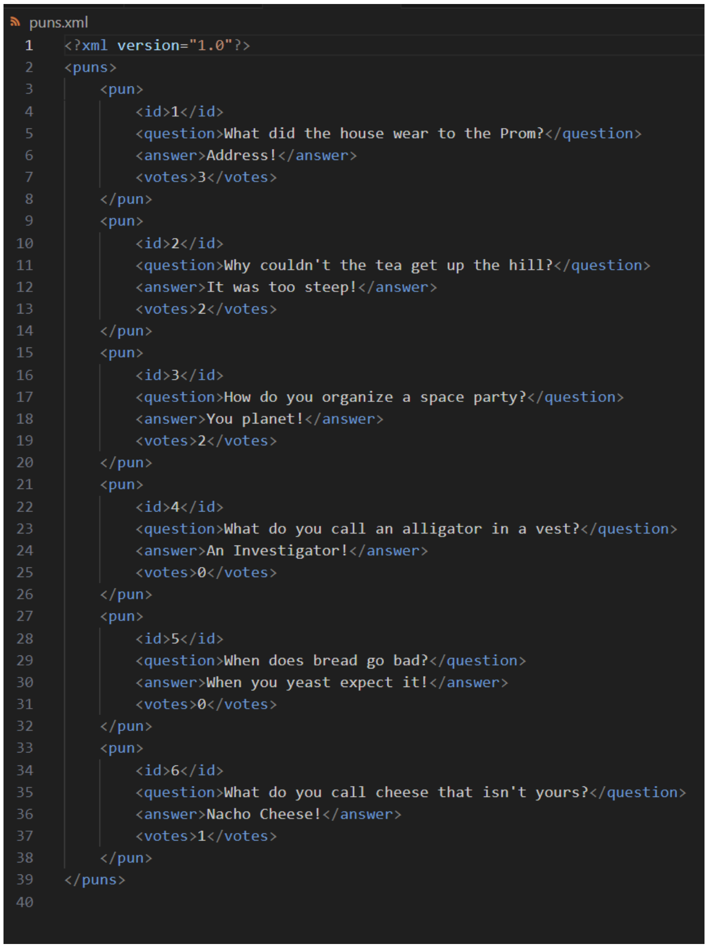
```javascript
$('#punList').hide();
//Function that asynchronously gets the data from puns.xml - this is what index.php calls upon button click - since the votes are NOT displayed
//jQuery and AJAX are used to asyncronously get the data from the xml page, and display it.
function loadPuns() {
  let counter = 0;
  $(document).ready(function () {
    $.ajax({
      url: 'puns.xml',
      dataType: 'xml',
      success: function (data) {
        $(data).find('pun').each(function () {
          let number = $(this).find('id').text();
          let question = $(this).find('question').text();
          let answer = $(this).find('answer').text();
          const btn = document.createElement('button');
          counter++;
          btn.innerText = "Vote For Pun " + counter;
          let listItem = $('<p id="pun">').html("Pun Number: " + number + "<br>" + question + ' . . .  ' + answer + '\n ');
          $('#punList').append(listItem);
          $('#vote').append(btn);
          $(btn).attr("name", counter);

        });
      }
    });
  });
  $('#view').hide();
};

//displayVotes is the AJAX request to get.php for the updated xml data
 const displayVotes = function() {
  let counter = 0;
  $(document).ready(function () {
    $.ajax({
      url: 'get.php',
      dataType: 'xml',
      success: function (data) {
        $(data).find('pun').each(function () {
          let number = $(this).find('id').text();
          let question = $(this).find('question').text();
          let answer = $(this).find('answer').text();
          let votes = $(this).find('votes').text();
          const btn = document.createElement('button');
          counter++;
          btn.innerText = "Vote For Pun " + counter;
          let listItem = $('<p id="pun2">').html(" Pun Number: " + number + "<br>" + question + ' . . .  ' + answer + '<p id=votes>Num of Votes ' + votes + '</p></h6> \n ');
          $('#punVotes').append(listItem);

        });
      }
    });
  });


};
```

Votes.php



```php
<!DOCTYPE html>
<html>
    <head>
        <!-- Making all the connections between my stylesheet, jquery and my javascript -->
        <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
        <script src="script.js"></script>
         <link rel="stylesheet" href="style.css"/>
    </head>

    <body>

        <ul>
        <li><a href="index.php">Home & Vote</a></li>
        <li><a href="votes.php">Standings</a></li>
        </ul>
         <h1>Which Pun Wins? - Current Votes</h1>
        <h5>Below shows the vote standings for each pun! <br> View the votes for each pun, see how your vote is reflected!</h5>
        <div id="stats">
         <!-- Button to trigger the php retrieval of the newly updated XML stats-->
         <br>
         <button id="punStat" onclick="displayVotes()">Show Votes</button>
         <div id='punVotes'></div>
        </div>
        <?php
        $file = file_get_contents('puns.xml');
        ?>

    </body>
</html>
```

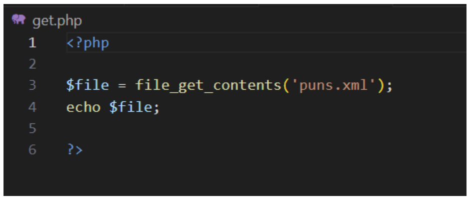
Puns.xml (the votes element is always changing through user interaction, therefore <votes> values may be different to what you see since I have been testing/will be testing using the browser).



puns.xml

```xml
<?xml version="1.0"?>
<puns>
    <pun>
        <id>1</id>
        <question>What did the house wear to the Prom?</question>
        <answer>Address!</answer>
        <votes>3</votes>
    </pun>
    <pun>
        <id>2</id>
        <question>Why couldn't the tea get up the hill?</question>
        <answer>It was too steep!</answer>
        <votes>2</votes>
    </pun>
    <pun>
        <id>3</id>
        <question>How do you organize a space party?</question>
        <answer>You planet!</answer>
        <votes>2</votes>
    </pun>
    <pun>
        <id>4</id>
        <question>What do you call an alligator in a vest?</question>
        <answer>An Investigator!</answer>
        <votes>0</votes>
    </pun>
    <pun>
        <id>5</id>
        <question>When does bread go bad?</question>
        <answer>When you yeast expect it!</answer>
        <votes>0</votes>
    </pun>
    <pun>
        <id>6</id>
        <question>What do you call cheese that isn't yours?</question>
        <answer>Nacho Cheese!</answer>
        <votes>1</votes>
    </pun>
</puns>

```

Get.php:



get.php

```php
<?php

$file = file_get_contents('puns.xml');
echo $file;

?>
```

What you want to accomplish with the site development why you chose this, and how it is supposed to work.

I created a site to show puns, and how different puns are rated, which allows users to view them and vote for their personal favorite. I chose to make this lighthearted website about jokes/puns because I love puns! The 6 puns that I included are some of my favorites, and always make me laugh. This website is meant to make others laugh as well, by sharing some of my favorite jokes. I find it interesting to see which one is the leading Pun, or which ones are the favorites so that I may know what kind of jokes people find the funniest, and entertaining. This can help me decide which jokes to tell others, based on user feedback, to get the best reaction! This website is supposed to work by allowing users to view the 6 different puns that I've included, read them through and decide which one they found funniest/enjoyed the most. Then, the website allows each user to vote for their favorite of the 6 puns included. Once they have voted, the site records which one is their favorite, and updates the XML document to increment the 'votes' element of the pun that they voted for. Then, if the user would like, the website allows them to view the current standings of each pun, highlighting how many votes each pun has (which also reflects their vote). Then, they may know which jokes other users are finding the funniest, and if the joke they voted for is one of the most popular, or not. This website allows users to embrace their humorous side, by exploring these jokes, and their competitive side, to see if the joke they have voted for is 'winning' in terms of most votes!

Short descriptions of the struggles you encountered.

- ❖ Including jQuery to make the AJAX request.
    - ➢ Including jQuery was a struggle that I needed to overcome. It was difficult at first to use jQuery because of its different syntax and rules. After additional research,

and looking at examples of jQuery syntax, I was able to understand it better and include it in my JavaScript. I am very happy that I was able to include jQuery because it makes writing the script so much more efficient and makes the source code much neater and more organized than raw JavaScript.

- Updating the ‘votes’ element in live time.
    - This was a struggle that really took me a long time to figure out. As seen in the website, after the user votes, it is important that their vote is immediately reflected in the XML document, so that they can view all the votes for each joke, including their own vote. At first, I wanted to incorporate AJAX functionality to retrieve and display the xml data, including the votes. However, shortly after I realized that the votes were not updating properly. I couldn’t understand why, being that the actual XML document was updating properly, but it was not being reflected in the browser. First, I decided to ditch the AJAX request, and just use php, however this did not accurately reflect the assignment criteria, since it is an AJAX focused assignment. After some in depth research, and help from professionals in the field, I was able to make the AJAX request for the updated values using the get.php file/technique seen in the source code above.

References

AJAX Introduction. (n.d.). W3Schools. Retrieved October 29, 2024, from https://www.w3schools.com/xml/ajax_intro.asp

A Beginner's Guide to the PHP SimpleXML Library for XML Parsing. (2023, April 21). Reintech. Retrieved October 29, 2024, from https://reintech.io/blog/beginners-guide-php-simplexml-library-xml-parsing

De Laurentiis, G. (n.d.). *XML Parser*. W3Schools. Retrieved October 29, 2024, from https://www.w3schools.com/xml/xml_parser.asp

Jain, S. (2024, May 2). *How to Access XML Data via JavaScript ?* GeeksforGeeks. Retrieved October 29, 2024, from https://www.geeksforgeeks.org/how-to-access-xml-data-via-javascript/

jQuery.getJSON(). (n.d.). jQuery API. Retrieved October 29, 2024, from https://api.jquery.com/jquery.getJSON/

Korobkov, B. (n.d.). *SimpleXML Functions - Manual*. PHP. Retrieved October 29, 2024, from https://www.php.net/manual/en/ref.simplexml.php

PHP Tutorial. (n.d.). W3Schools. Retrieved October 29, 2024, from https://www.w3schools.com/php/default.asp