PHP File IO\prog14.php

```php
<?php
// Set file directoy to secure directory used in hangman
$filesDir = '/app/secure_data/data/';

// Create directory if it doesn't exist
if (!is_dir($filesDir)) {
    mkdir($filesDir, 0755, true);
}

// Create a text file array with proper permissions
// Had some issues due to my docker php setup but working now
$files = [
    'prime' => [
        'path' => $filesDir . 'prime.txt',
        'permissions' => 0755
    ],
    'armstrong' => [
        'path' => $filesDir . 'armstrong.txt',
        'permissions' => 0755
    ],
    'fibonacci' => [
        'path' => $filesDir . 'fibonacci.txt',
        'permissions' => 0755
    ],
    'none' => [
        'path' => $filesDir . 'none.txt',
        'permissions' => 0755
    ]
];

// Check for first time user and create empty files if cookie is empty
if (!isset($_COOKIE['Active'])) {
    // For each of the files in the array
    foreach ($files as $fileNameKey => $fileInfo) {
        // Set the path
        $filePath = $fileInfo['path'];
```

```php
        // Set the permissions
        $filePermissions = $fileInfo['permissions'];
        // Attempt to create the empty file
        if (file_put_contents($filePath, '') === false) {
            // Case if it failed to make file
            //echo "Error: Could not create file " . $filePath . "\n";
        } else {
            // Case if it succeeds to create file
            //echo "Successfully created file: " . $filePath . "\n";
            // Attempt to set the permissions of the file
            if (!chmod($filePath, $filePermissions)) {
                // Case if it could not set permissions
                //echo "Error: Could not set permissions for " . $filePath . "\n";
            } else {
                // Case if it does set the permissions
                //echo "Successfully set permissions for: " . $filePath . " to " . decoct($filePermissions) . "\n";
            }
        }
    }
    // After files are created, set the cookie in the browser for 1 year expiration
    setcookie('Active', 'True', time() + (365 * 24 * 60 * 60));
}

// Helper function so see if a number is armstrong
function isArmstrong($num) {
    // Return false if it is a single digit number as these are not considered
    if ($num >= 0 && $num < 10) {
        return false;
    }
    // Setup variable sum
    $sum = 0;
    // Digits takes the input, converts to a string, and puts the characters in an array digits
    $digits = str_split((string)$num);
    // Set the power to be exponentiated
    $power = count($digits);
    // To calculate armstrong, raise each number to the power and sum them together
    foreach ($digits as $digit) {
        $sum += pow((int)$digit, $power);
```

```php
    }
    // True if the sum and number are equal
    return $sum == $num;
}

// Helper function to calcualate fibonacci
function isFibonacci($num) {
    // Base cases for the Fibonacci sequence
    if ($num == 0 || $num == 1) {
        return true;
    }
    $a = 0; // Represents F(n-2)
    $b = 1; // Represents F(n-1)

    // Generate Fibonacci numbers until $num
    while (true) {
        // Calculate the next Fibonacci number
        $nextFib = $a + $b;
        // Compare the next fib to the current num
        if ($nextFib == $num) {
            return true;
        }
        // If the next Fibonacci number is greater than $num, then return false
        if ($nextFib > $num) {
            return false;
        }
        // Update $a and $b for the next iteration
        $a = $b;
        $b = $nextFib;
    }
}

// Helper function to check if a number is prime
function isPrime($num) {
    // If 2, return false
    if ($num < 2) {
        return false;
    }
```

```php
    // Start by dividing each number by 2 and increment
    for ($i = 2; $i <= sqrt($num); $i++) {
        // If there is no remainder, then it is not a prime number
        if ($num % $i == 0) return false;
    }
    return true;
}

// Helper function to add numbers to their respected text file
function appendNumber($file, $num) {
    // Append number to file with a new line
    file_put_contents($file, $num . "\n", FILE_APPEND);
}

// Helper function to read numbers from the file
function readNumbers($file) {
    // Ensure $file is valid and exist
    if (!is_string($file) || !file_exists($file)) {
        return [];
    }
    // Pull the content from the file and remove whitespace with trim
    $content = trim(file_get_contents($file));
    // Check if the content is empty after trimming, return empty array
    if (empty($content)) {
        return [];
    }
    // Split the content into an array of lines
    $lines = explode("\n", $content);
    // Filter the array to keep only the numbers
    $numericLines = array_filter($lines, 'is_numeric');
    return $numericLines;
}

// Path handling

// Handle requests by checking if action is set in request
$action = isset($_GET['action']) ? $_GET['action'] : '';

```

```php
// If the operation is post and the action is check
if ($_SERVER['REQUEST_METHOD'] === 'POST' && $action === 'check') {
    // Retrieve the numbers sent from the js
    $numbers = isset($_POST['numbers']) ? $_POST['numbers'] : '';
    // Take the numbers and put them in an array by removing , whitespace and filtering for numbers only
    $numbers = array_filter(array_map('trim', explode(',', $numbers)), 'is_numeric');
    // Empties out the files just in case they have info
    foreach ($files as $fileInfo) {
        file_put_contents($fileInfo['path'], '');
    }
    // For each of the numbers given
    foreach ($numbers as $num) {
        // Ensure they are converted to int types
        $num = (int)$num;
        // Helper variable to send the number to the none list if it is not a part of any other list
        $foundCategory = false;

        // Check for each of the types
        if (isArmstrong($num)) {
            appendNumber($files['armstrong']['path'], $num);
            $foundCategory = true;
        }
        if (isFibonacci($num)) {
            appendNumber($files['fibonacci']['path'], $num);
            $foundCategory = true;
        }
        if (isPrime($num)) {
            appendNumber($files['prime']['path'], $num);
            $foundCategory = true;
        }
        // If the number does not belong to any category then place it in none
        if (!$foundCategory) {
            appendNumber($files['none']['path'], $num);
        }
    }
    // Display success message
    echo "<p>Numbers processed successfully.</p>";
    exit;
```

```php
}

// If the armstrong numbers are requested
if ($action === 'armstrong') {
    // Set the header
    header('Content-Type: application/json');
    // Echo the numbers read in from the text file in json
    echo json_encode(readNumbers($files['armstrong']['path']));
    exit;
}

// If the fibonacci numbers are requested
if ($action === 'fibonacci') {
    header('Content-Type: application/json');
    echo json_encode(readNumbers($files['fibonacci']['path']));
    exit;
}

// If the prime numbers are requested
if ($action === 'prime') {
    header('Content-Type: application/json');
    echo json_encode(readNumbers($files['prime']['path']));
    exit;

}

// If the none type numbers are requested
if ($action === 'none') {
    header('Content-Type: application/json');
    echo json_encode(readNumbers($files['none']['path']));
    exit;
}

// Function to handle reset request
if ($action === 'reset') {
    // For each of the files in the file list
    foreach ($files as $fileKey => $fileInfo) {
        // Grab the path
```

```php
        $filePath = $fileInfo['path'];
        // Check to see if the file exist
        if (file_exists($filePath)) {
            // Delete the file
            if (!unlink($filePath)) {
                // Case if operation fails
                error_log("Error: Could not delete file " . $filePath);
            } else {
                // Case if success
                error_log("Successfully deleted file: " . $filePath);
            }
        }
    }
    // Delete the cookie by setting its expiration to a past time
    setcookie('Active', '', time() - 3600); // Delete cookie
    // Echo the reset success
    echo "<p>Files and cookie reset successfully.</p>";
    exit;
}
?>
```