

Sean Farrell

7/12/2025

CPSC 4750 Wooster

PHP File I/O App

```

1  <!--
2  Sean Farrell
3  7/12/2025
4  CPSC 3750 Wooster
5  ASSIGNMENT: PHP File I/O App
6  DESCRIPTION: This PHP application allows users to input numbers and categorize them as prime, Armstrong, Fibonacci, or none.
7  It uses a form to submit the numbers and displays the results dynamically.
8  THIS FILE: Logic.php contains the serverside php logic for finding numbers that are or not armstrong, primer of fibonacci.
9  -->
10 <?php
11 // Check if the cookie exists (first time user)
12 if (!isset($_COOKIE['number_categorizer_visited'])) {
13     // Set the cookie for 30 days
14     setcookie('number_categorizer_visited', 'true', time() + (86400 * 30), "/");
15 }
16 // Create the four empty text files
17 $files = ['prime.txt', 'armstrong.txt', 'fibonacci.txt', 'none.txt'];
18 foreach ($files as $file) {
19     file_put_contents($file, '');
20 }
21 }
22 }
23 // Function to check if a number is prime
24 function isPrime($num) {
25     if ($num <= 1) return false;
26     if ($num <= 3) return true;
27     if ($num % 2 == 0 || $num % 3 == 0) return false;
28 }
29 $i = 5;
30 while ($i * $i <= $num) {
31     if ($num % $i == 0 || $num % ($i + 2) == 0) return false;
32     $i += 6;
33 }
34 return true;
35 }
36 }
37 // Function to check if a number is an Armstrong number
38 function isArmstrong($num) {
39     $sum = 0;
40     $temp = $num;
41     $digits = strlen((string)$num);
42 }
43 while ($temp > 0) {
44     $remainder = $temp % 10;
45     $sum += pow($remainder, $digits);
46     $temp = (int)($temp / 10);
47 }
48 return $sum == $num;
49 }
50 }
51 }
52 // Function to check if a number is a Fibonacci number
53 function isFibonacci($num) {
54     // A number is Fibonacci if and only if one or both of (5*n^2 + 4) or (5*n^2 - 4) is a perfect square
55     return isPerfectSquare(5 * $num * $num + 4) || isPerfectSquare(5 * $num * $num - 4);
56 }
57 }
58 function isPerfectSquare($num) {
59     $sqrt = (int)sqrt($num);
60     return ($sqrt * $sqrt == $num);
61 }
62 }
63 // Function to append a number to a file if it doesn't already exist
64 function appendToFile($file, $number) {
65     $content = file_get_contents($file);
66     $numbers = explode(',', $content);
67     $numbers = array_filter($numbers); // Remove empty values
68 }
69 if (!in_array($number, $numbers)) {
70     if (!empty($content)) {
71         $content .= ',';
72     }
73     $content .= $number;
74     file_put_contents($file, $content);
75 }
76 }
77 }
78 // Function to display numbers from a file
79 function displayNumbers($file, $type) {
80     $content = file_get_contents($file);
81     $numbers = explode(',', $content);
82     $numbers = array_filter($numbers); // Remove empty values
83 }
84 $html = "<h2>$type Numbers</h2>";
85 }
86 if (empty($numbers)) {
87     $html .= "<p>No $type numbers found.</p>";
88 } else {
89     foreach ($numbers as $number) {
90         $html .= "<div class='number-item'>$number</div>";
91     }
92 }
93 return $html;
94 }
95 }
96 }
97 $result_message = '';
98 }
99 // Check which action was requested
100 if (isset($_POST['action'])) {
101     $action = $_POST['action'];
102 }
103 switch ($action) {
104     case 'check':
105         // Process the numbers entered by the user
106         if (isset($_POST['numbers']) && !empty(trim($_POST['numbers']))) {
107             $input = $_POST['numbers'];
108             // Extract numbers (separated by spaces, commas, or new lines)
109             preg_match_all('/\d+/', $input, $matches);
110             $numbers = $matches[0];
111 }
112 foreach ($numbers as $number) {
113     if (isPrime($number)) {
114         appendToFile('prime.txt', $number);
115     }
116     if (isArmstrong($number)) {
117         appendToFile('armstrong.txt', $number);

```

```

62 // Function to append a number to a file if it doesn't already exist
63 function appendToFile($file, $number) {
64     $content = file_get_contents($file);
65     $numbers = explode(',', $content);
66     $numbers = array_filter($numbers); // Remove empty values
67
68     if (!in_array($number, $numbers)) {
69         if (!empty($content)) {
70             $content .= ',';
71         }
72         $content .= $number;
73         file_put_contents($file, $content);
74     }
75 }
76
77 // Function to display numbers from a file
78 function displayNumbers($file, $type) {
79     $content = file_get_contents($file);
80     $numbers = explode(',', $content);
81     $numbers = array_filter($numbers); // Remove empty values
82
83     $html = "<h2>$type Numbers</h2>";
84
85     if (empty($numbers)) {
86         $html .= "<p>No $type numbers found.</p>";
87     } else {
88         foreach ($numbers as $number) {
89             $html .= "<div class='number-item'>$number</div>";
90         }
91     }
92
93     return $html;
94 }
95
96 $result_message = '';
97
98 // Check which action was requested
99 if (isset($_POST['action'])) {
100     $action = $_POST['action'];
101
102     switch ($action) {
103         case 'check':
104             // Process the numbers entered by the user
105             if (isset($_POST['numbers']) && !empty(trim($_POST['numbers']))) {
106                 $input = $_POST['numbers'];
107                 // Extract numbers (separated by spaces, commas, or new lines)
108                 preg_match_all('/\d+/', $input, $matches);
109                 $numbers = $matches[0];
110
111                 foreach ($numbers as $number) {
112                     if (isPrime($number)) {
113                         appendToFile('prime.txt', $number);
114                     }
115                     if (isArmstrong($number)) {
116                         appendToFile('armstrong.txt', $number);
117                     }
118                     if (isFibonacci($number)) {
119                         appendToFile('fibonacci.txt', $number);
120                     }
121                     if (!isPrime($number) && !isArmstrong($number) && !isFibonacci($number)) {
122                         appendToFile('none.txt', $number);
123                     }
124                 }
125
126                 $result_message = "<h2>Numbers Processed</h2>";
127                 $result_message .= "<p>Your numbers have been processed and categorized.</p>";
128             } else {
129                 $result_message = "<h2>Error</h2>";
130                 $result_message .= "<p>Please enter some numbers to check.</p>";
131             }
132             break;
133
134         case 'prime':
135             $result_message = displayNumbers('prime.txt', 'Prime');
136             break;
137
138         case 'armstrong':
139             $result_message = displayNumbers('armstrong.txt', 'Armstrong');
140             break;
141
142         case 'fibonacci':
143             $result_message = displayNumbers('fibonacci.txt', 'Fibonacci');
144             break;
145
146         case 'none':
147             $result_message = displayNumbers('none.txt', 'None');
148             break;
149
150         case 'reset':
151             // Delete all files and clear the cookie
152             $files = ['prime.txt', 'armstrong.txt', 'fibonacci.txt', 'none.txt'];
153             foreach ($files as $file) {
154                 file_put_contents($file, '');
155             }
156             setcookie('number_categorizer_visited', '', time() - 3600, "/");
157
158             $result_message = "<h2>Reset Complete</h2>";
159             $result_message .= "<p>All files have been reset and the cookie has been cleared.</p>";
160             break;
161     }
162 }
163
164 // Always include the HTML file
165 include 'PHP10.html';
166
167 }
168

```

```
1
2 <!--
3 Sean Farrell
4 7/12/2025
5 CPSC 3750 Wooster
6 ASSIGNMENT: PHP File I/O App
7 DESCRIPTION: This PHP application allows users to input numbers and categorize them as prime, Armstrong, Fibonacci, or none.
8 It uses a form to submit the numbers and displays the results dynamically.
9 THIS FILE: main.php initiates the application and includes the logic.php for processing in the website.
10 <?php
11 // Include the processing logic
12 require_once 'logic.php';
13 ?>
```