

```

1 <html>
2   <head>
3     <meta charset='utf-8'>
4     <meta http-equiv='X-UA-Compatible' content='IE=edge'>
5     <title>yo</title>
6     <meta name='viewport' content='width=device-width, initial-scale=1'>
7     <link rel='stylesheet' type='text/css' media='screen' href='results.css'>
8     <link rel="preconnect" href="https://fonts.gstatic.com">
9     <link href="https://fonts.googleapis.com/css2?
10 family=Josefin+Sans&display=swap" rel="stylesheet">
11     <script type="text/javascript"
12 src="//ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
13     <script src='main.js'></script>
14   </head>
15   <body>
16     <div class="topnav">
17       <a href="main.html">Home</a>
18       <a class="active" href="select.php">Visualize (profilin)</a>
19       <a href="select_ns1tp.php">Visualize (nsLTP)</a>
20       <a href="select_pr10.php">Visualize (pr10)</a>
21       <a href="personal.php">Inspect</a>
22     </div>
23
24     <style>
25       body{
26         font-family: 'Josefin Sans', sans-serif;
27       }
28     </style>
29     <?php
30     $servername = "localhost";
31     $username = "root";
32     $password = "Iatwbofm21!";
33     $dbname = "crossreactivity";
34
35     $temp = $_GET['display'];
36     # echo "****" . $temp;
37     #echo "<br>";
38     // Create connection
39     $conn = new mysqli($servername, $username, $password, $dbname);
40     // Check connection
41     if ($conn->connect_error) {
42       die("Connection failed: " . $conn->connect_error);
43     }
44
45     $sql = "SELECT name, origin FROM crossreactivity";
46     #echo $sql;
47     $result = $conn->query($sql);
48     if ($result->num_rows > 0) {
49       // output data of each row
50       $maps = array();
51       $maps_otn = array();
52       while($row = $result->fetch_assoc()) {
53         $maps[$row['name']] = $row['origin'];
54         $maps_otn[$row['origin']] = $row['name'];
55       }
56     } else {
57       echo "0 results";
58     }
59
60
61     $query = "SELECT * FROM crossreactivity WHERE name = '". $temp. "'";
62     #echo "<br>";
63     #echo $query;
64     #echo "<br>";
65     $result = mysqli_query($conn, $query);
66
67     $rows = mysqli_fetch_all ($result, MYSQLI_ASSOC);
68
69     foreach( $rows as &$row){
70       array_multisort($row, SORT_DESC, SORT_NUMERIC);

```

```

71     $origin = $row['origin'];
72     $name = $row['name'];
73
74
75     echo "<h1><center>$origin</center></h1>";
76     echo "<br>";
77     #echo $name;
78
79     unset($row['origin']);
80     unset($row['name']);
81
82     /*
83     foreach($row as $key => $value){
84         echo $value;
85         echo " ";
86     }
87     */
88
89     echo "<br>";
90
91     $last = end($row);
92     $first = reset($row);
93
94     #echo $row;
95
96     foreach($row as $key => $value){
97         #echo $value;
98         #echo $maps[$key]." : ".$key . " : " . $value . "<br>";
99         $last = (float)$last ;
100         $color = numberToColor($value,0,100, ['#8de667', '#fff5f5',
101 '#CC0000']);
102         #echo $color;
103         echo "<div class='flip-card'>
104             <div class='flip-card-inner'>
105                 <div class='flip-card-front'>
106                     <div style=' padding:10%;display:inline-block;
107 height:80%;width:75%; font-size:28px; color: black; background-color:" . $color . ";
108 text-align:center'>$maps[$key]</div>
109                 </div>
110                 <div class='flip-card-back'>
111                     <h1>$maps[$key]</h1>
112                     <p>$key</p>
113                     <p>$value</p>
114                 </div>
115             </div>
116             #echo "<div class = 'result'>$final</div>";
117         }
118         echo "<br>";
119         echo "<br>";
120         echo "<br>";
121         echo "<br>";
122     }
123     $conn->close();
124
125     function numberToColor($value, $min, $max, $gradientColors = null)
126     {
127         // Ensure value is in range
128         if ($value < $min) {
129             $value = $min;
130         }
131         if ($value > $max) {
132             $value = $max;
133         }
134
135         // Normalize min-max range to [0, positive_value]
136         $max -= $min;
137         $value -= $min;
138         $min = 0;
139
140         // Calculate distance from min to max in [0,1]

```

```

140     $distFromMin = $value / $max;
141
142     // Define start and end color
143     if (count($gradientColors) == 0) {
144         return numberToColor($value, $min, $max, ['#CC0000', '#EEEE00',
145 '#00FF00']);
146     } else if (count($gradientColors) == 2) {
147         $startColor = $gradientColors[0];
148         $endColor = $gradientColors[1];
149     } else if (count($gradientColors) > 2) {
150         $startColor = $gradientColors[floor($distFromMin *
151 (count($gradientColors) - 1))];
152         $endColor = $gradientColors[ceil($distFromMin *
153 (count($gradientColors) - 1))];
154
155         $distFromMin *= count($gradientColors) - 1;
156         while ($distFromMin > 1) {
157             $distFromMin--;
158         }
159     } else {
160         die("Please pass more than one color or null to use default red-green
161 colors.");
162     }
163
164     // Remove hex from string
165     if ($startColor[0] === '#') {
166         $startColor = substr($startColor, 1);
167     }
168     if ($endColor[0] === '#') {
169         $endColor = substr($endColor, 1);
170     }
171
172     // Parse hex
173     list($ra, $ga, $ba) = sscanf("#$startColor", "%02x%02x%02x");
174     list($rz, $gz, $bz) = sscanf("#$endColor", "%02x%02x%02x");
175
176     // Get rgb based on
177     $distFromMin = $distFromMin;
178     $distDiff = 1 - $distFromMin;
179     $r = intval(($rz * $distFromMin) + ($ra * $distDiff));
180     $r = min(max(0, $r), 255);
181     $g = intval(($gz * $distFromMin) + ($ga * $distDiff));
182     $g = min(max(0, $g), 255);
183     $b = intval(($bz * $distFromMin) + ($ba * $distDiff));
184     $b = min(max(0, $b), 255);
185
186     // Convert rgb back to hex
187     $rgbColorAsHex = '#' .
188         str_pad(dechex($r), 2, "0", STR_PAD_LEFT) .
189         str_pad(dechex($g), 2, "0", STR_PAD_LEFT) .
190         str_pad(dechex($b), 2, "0", STR_PAD_LEFT);
191
192     return $rgbColorAsHex;
193 }
194
195 ?>
196
197 
198 <p>The darker the red, the more you should avoid that substance, the lighter
199 the green, the more ok it is to be around. Flip the cards over to see a risk-score
200 between 0 and 100</p>
201 <br>
202 </body>
203 <script type = "text/javascript">
204     $( "img" ).each( function() {
205         var $img = $( this );
206         $img.width( $img.width() * .5 );
207     });
208 </script>
209 </html>

```