```

<html>
    <head>
        <meta charset='utf-8'>
        <meta http-equiv='X-UA-Compatible' content='IE=edge'>
        <title>yo</title>
        <meta name='viewport' content='width=device-width, initial-scale=1'>
        <link rel='stylesheet' type='text/css' media='screen' href='results.css'>
        <link rel="preconnect" href="https://fonts.gstatic.com">
        <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans&display=swap" rel="stylesheet">
        <script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
        <script src='main.js'></script>
    </head>
    <body>
        <div class="topnav">
            <a href="main.html">Home</a>
            <a class="active" href="select.php">Visualize (profilin)</a>
            <a href="select_nsltp.php">Visualize (nsLTP)</a>
            <a href="select_pr10.php">Visualize (pr10)</a>
            <a href="personal.php">Inspect</a>
        </div>

        <style>
            body{
                font-family: 'Josefin Sans', sans-serif;

            }
        </style>
        <?php
        $servername = "localhost";
        $username = "root";
        $password = "Iatwbofm21!";
        $dbname = "crossreactivity";

        $temp = $_GET['display'];
       # echo "***" .$temp;
        #echo "<br>";
        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
        }

        $sql = "SELECT name, origin FROM crossreactivity";
        #echo $sql;
        $result = $conn->query($sql);
        if ($result->num_rows > 0) {
            // output data of each row
            $maps = array();
            $maps_otn = array();
            while($row = $result->fetch_assoc()) {
                    $maps[$row['name']] = $row['origin'];
                    $maps_otn[$row['origin']] = $row['name'];
            }
            } else {
            echo "0 results";
        }


        $query = "SELECT * FROM crossreactivity WHERE name = '".$temp."'";
        #echo "<br>";
        #echo $query;
        #echo"<br>";
        $result = mysqli_query($conn, $query);

        $rows = mysqli_fetch_all ($result, MYSQLI_ASSOC);

        foreach( $rows as &$row){
            array_multisort($row, SORT_DESC, SORT_NUMERIC);

            $origin = $row['origin'];
            $name = $row['name'];

            echo "<h1><center>$origin</center></h1>";
            echo "<br>";
            #echo $name;

            unset($row['origin']);
            unset($row['name']);

            /*
            foreach($row as $key => $value){
                echo $value;
                echo " ";
            }
            */

            echo "<br>";

            $last = end($row);
            $first = reset($row);

            #echo $row;

            foreach($row as $key => $value){
                #echo $value;
                #echo $maps[$key]." : ".$key . " : " . $value . "<br>";
                $last = (float)$last ;
                $color = numberToColor($value,0,100, ['#8de667', '#fff5f5',
'#CC0000']);
                #echo $color;
                echo "<div class='flip-card'>
                        <div class='flip-card-inner'>
                            <div class='flip-card-front'>
                                <div style=' padding:10%;display:inline-block;
height:80%;width:75%; font-size:28px; color: black; background-color:" . $color . ";
text-align:center'>$maps[$key]</div>
                            </div>

                            <div class='flip-card-back'>
                                <h1>$maps[$key]</h1>
                                <p>$key</p>
                                <p>$value</p>
                            </div>
                        </div>
                    </div>";
                #echo "<div class = 'result'>$final</div>";
            }
            echo "<br>";
            echo "<br>";
            echo "<br>";
            echo "<br>";
        }
        $conn->close();

        function numberToColor($value, $min, $max, $gradientColors = null)
        {
            // Ensure value is in range
            if ($value < $min) {
                $value = $min;
            }
            if ($value > $max) {
                $value = $max;
            }

            // Normalize min-max range to [0, positive_value]
            $max -= $min;
            $value -= $min;
            $min = 0;

            // Calculate distance from min to max in [0,1]
            $distFromMin = $value / $max;

            // Define start and end color
            if (count($gradientColors) == 0) {
                return numberToColor($value, $min, $max, ['#CC0000', '#EEEE00', '#00FF00']);
            } else if (count($gradientColors) == 2) {
                $startColor = $gradientColors[0];
                $endColor = $gradientColors[1];
            } else if (count($gradientColors) > 2) {
                $startColor = $gradientColors[floor($distFromMin * (count($gradientColors) - 1))];
                $endColor = $gradientColors[ceil($distFromMin * (count($gradientColors) - 1))];

                $distFromMin *= count($gradientColors) - 1;
                while ($distFromMin > 1) {
                    $distFromMin--;
                }
            } else {
                die("Please pass more than one color or null to use default red-green colors.");
            }

            // Remove hex from string
            if ($startColor[0] === '#') {
                $startColor = substr($startColor, 1);
            }
            if ($endColor[0] === '#') {
                $endColor = substr($endColor, 1);
            }

            // Parse hex
            list($ra, $ga, $ba) = sscanf("#$startColor", "#%02x%02x%02x");
            list($rz, $gz, $bz) = sscanf("#$endColor", "#%02x%02x%02x");

            // Get rgb based on
            $distFromMin = $distFromMin;
            $distDiff = 1 - $distFromMin;
            $r = intval(($rz * $distFromMin) + ($ra * $distDiff));
            $r = min(max(0, $r), 255);
            $g = intval(($gz * $distFromMin) + ($ga * $distDiff));
            $g = min(max(0, $g), 255);
            $b = intval(($bz * $distFromMin) + ($ba * $distDiff));
            $b = min(max(0, $b), 255);

            // Convert rgb back to hex
            $rgbColorAsHex = '#' .
                str_pad(dechex($r), 2, "0", STR_PAD_LEFT) .
                str_pad(dechex($g), 2, "0", STR_PAD_LEFT) .
                str_pad(dechex($b), 2, "0", STR_PAD_LEFT);

            return $rgbColorAsHex;
        }

        ?>

        <img src="key2.png">
        <p>The darker the red, the more you should avoid that substance, the lighter the green, the more ok it is to be around. Flip the cards over to see a risk-score between 0 and 100</p>
        <br>
    </body>
    <script type = "text/javascript">
        $( "img" ).each( function() {
            var $img = $( this );
            $img.width( $img.width() * .5 );
        });
    </script>
</html>

```