```
<!DOCTYPE html>
<html>
<?php
/*
 * Author : Lucien Camuglia
 * Description : index page
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */
include './Includes/header.php'
?>

    <body>
        <?php
        DisplayMenu(basename($_SERVER['PHP_SELF']));
        ?>
        <?php include './Includes/footer.php'; ?>

        <div class="col-lg-6 col-lg-offset-3">
        <p>The website was made for my Technician diploma, it allows sharing itineraries between motorcycle enthusiasts. The site is set-up with HTML/PHP/CSS/AJAX and Google’s API. It allows users to create trips, to share and modify them.</p>


        <p>Different criteria can be defined by the user to create the itinerary:</p>
        <ul>
            <li> Travel Time </li>
            <li> Type of road </li>
            <li> Change in altitude, etc.</li>
        </ul>



        <p>This website gives the user diverse informations. One example could be the theoretical consumption of their motorcycle for the ride or travel time. The rider can also upload or download files directly from his GPS.</p>
        </div>
    </body>

</html>

```

```
<!DOCTYPE html>
<?php
/*
 * Author : Lucien Camuglia
 * Description : Page with all itinaries
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */?>
<html>
    <head>
        <?php include './Includes/header.php' ?>
        <title></title>
    </head>
    <body style="height: 500px;" >
        <?php
        DisplayMenu(basename($_SERVER['PHP_SELF']));
        ?>
        <script src="./JS/Map.js"></script>
        <script>

        </script>
        <div class="col-sm-1 "><h2>Filters  :</h2>
            <h6 class="col-xs-9">Enabled</h6> <input class="col-sm-3 checkbox-inline" type="checkbox" onclick="EnabledDisabledFilters(this)"/>
            <div id="filters" hidden="true">
                <p><label>Sinuosity</label><input id="sinuosity" type="range" min="<?php echo GetLessSinuousRoad(); ?>" max="<?php echo GetMostSinuousRoad() + 1; ?>" /></p>
                <p><label>Slope</label><input id="slope" type="range"  min="<?php echo GetLessSteepestRoad(); ?>" max="<?php echo GetMostSteepestRoad(); ?>" /></p>
                <p><label>Highway </label><input id="highway" class="col-sm-3 checkbox-inline"  type="checkbox"  /></p>

            </div>
        </div>
        <div class="routes col-lg-2 " style="height: 110%">
            <?php
            $routes = GetRoutes();
            foreach ($routes as $route) {
                echo"<div class=\"route\" name=\"" . $route["idRoute"] . "\">" . $route["RouteName"] . "<span class=\"by\"> by " . $route["Username"] . "</span></div>";
            }
            ?>

        </div>
        <input id="pac-input" class="controls" type="text" placeholder="Search Box" hidden>
        <div id="map" class="col-lg-8" style="height: 110%;"></div>
        <div><button class="btn btn-primary" onclick="DownloadRoute();" >Download route</button></div>

        <div class="col-lg-5 col-lg-offset-4 routeControl hidden " >
            <center><h2>Road Infos</h2></center>
            <h4 class="col-lg-12" id="InfoHighway">This road does not includes highways</h4>
            <p><span style="text-align: left;" class="col-lg-6" id="InfoLength">Length : 12km</span><span style="text-align: right;" class="col-lg-6" id="InfoDuration">Duration : 0h30min</span></p>
            <p><span style="text-align: left;" class="col-lg-6" id="InfoSinuosity">Sinuosity : 0/10</span><span style="text-align: right;" class="col-lg-6" id="InfoSlope">Slope : 0/10</span></p>
            <p><span style="text-align: left;" class="col-lg-8" id="InfoConsumption">Consumption : 52 liters</span><span style="text-align: right;" class="col-lg-6"></span></p>

        </div>
          <?php include './Includes/footer.php'; ?>
    </body>
</html>

```

```
<!DOCTYPE html>
<?php
/*
 * Author : Lucien Camuglia
 * Description : User page for own itinaries
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */
?>
<html>
    <head>
        <?php include './Includes/header.php'; ?>
        <title></title>
    </head>
    <body style="height: 500px;" >
        <?php
        DisplayMenu(basename($_SERVER['PHP_SELF']));
        importerTrajetModal();
        newRouteModal();
        if (isset($_GET["NewIti"])) {
            $idRoute = $_GET["id"];
        }
        ?>
        <script src="./JS/Map.js"></script>
        <script>
<?php
if (isset($idRoute)) {
    ?>
                var Points = LoadPoints(<?php echo $idRoute ?>);
                console.log(Points);
                Points = SnappPoints2Road(Points);
                SaveNewLocation(<?php echo $idRoute ?>, Points);
    <?php
}
?>
        </script>
        <div class=" routes col-lg-offset-1 col-lg-2" style="height: 500px;">

            <?php
            $routes = GetRoutes($_SESSION["id"]);
            foreach ($routes as $route) {
                echo"<div class=\"route\" name=\"" . $route["idRoute"] . "\">" . $route["RouteName"] . "</div>";
            }
            ?>


        </div>
        <input id="pac-input" class="controls" type="text" placeholder="Search Box" hidden/>
        <div id="map" class="col-lg-7" style="height: 500px;"></div>

        <div class="btnZone col-lg-2 top-buffer"  >
            <!-- <button id="btnModif"  class="hidden btn btn-primary btn-block" name="none">Modifier le trajet</button>-->

            <label class="checkbox-inline SaveRouteControls hidden">
```

```
                <input id="chkHighway" type="checkbox" checked data-toggle="toggle"> Highways
            </label>

            <button id="SaveRoute" class=" btn btn-primary btn-block SaveRouteControls hidden" data-target="#newRoutemodal" data-toggle="modal">Save the route</button>
            <button id="ClearRoute" class=" btn btn-primary btn-block SaveRouteControls hidden">Clear the route</button>

            <button class="btn btn-block" id="StartCreation">Créer un nouveau trajet</button>
            <button class="btn  btn-block" data-toggle="modal" data-target="#modalimport" onclick='$("#msgCont").attr("Hidden", "true")' >Importer un nouveau trajet</button>

            <button id="RefreshRoute" class=" btn btn-primary btn-block  hidden modifRoute">Refresh the route</button>
            <button id="DeleteRoute" class=" btn btn-danger btn-block  hidden routeControl">Delete the route</button>

        </div>

        <div id="pointsInfo" class="col-lg-6 col-lg-offset-3 hidden modifRoute"  >

            <div id="pointStart">
                <div class="Pointname col-lg-12" >
                    <span class="col-lg-2 align-middle" style="height: 100px;" >Start : </span>
                    <div class="col-lg-10" >
                        <div>
                            <label class="col-sm-3">Latitude : </label> <input id="inpLatPoint0" type="text" value="0.000">
                        </div>
                        <div>
                            <label class="col-sm-3">Longitude : </label> <input id="inpLonPoint0" type="text" value="0.000">
                        </div>
                    </div>
                </div>

            </div>

            <div id="pointEnd">
                <div class="Pointname col-lg-12" >
                    <span class="col-lg-2 align-middle" style="height: 100%;" >End : </span>
                    <div class="col-lg-10" >
                        <div>
                            <label class="col-sm-3">Latitude : </label> <input id="inpLatPoint1"  type="text" value="0.000">
                        </div>
                        <div>
                            <label class="col-sm-3">Longitude : </label> <input id="inpLonPoint1"  type="text" value="0.000">
                        </div>
                    </div>
                </div>
            </div>

        </div>

        <div class="modalLoad"></div>

        <div class="col-lg-5 col-lg-offset-4 routeControl hidden" >
            <center><h2>Road Infos</h2></center>
            <h4 class="col-lg-12" id="InfoHighway">This road does not includes highways</h4>
            <p><span style="text-align: left;" class="col-lg-6" id="InfoLength">Length : 12km</span><span style="text-align: right;" class="col-lg-6" id="InfoDuration">Duration : 0h30min</span></p>
            <p><span style="text-align: left;" class="col-lg-6" id="InfoSinuosity">Sinuosity : 0/10</span><span style="text-align: right;" class="col-lg-6" id="InfoSlope">Slope : 0/10</span></p>
            <p><span style="text-align: left;" class="col-lg-8" id="InfoConsumption">Consumption : 52 liters</span><span style="text-align: right;" class="col-lg-6"></span></p>

        </div>
        <?php include './Includes/footer.php'; ?>
    </body>
</html>

```

```
<!DOCTYPE html>
<?php
/*
 * Author : Lucien Camuglia
 * Description : Managing users page
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */
?>

<html>
    <head>
        <?php include './Includes/header.php' ?>
        <script src="./JS/ManageUsers.js"></script>
        <title></title>
    </head>
    <body>
        <?php
        DisplayMenu(basename($_SERVER['PHP_SELF']));
        ?>

        <table class="table table-striped table-bordered table-responsive" cellspacing="0" id="">
            <thead>
                <tr>
                    <th rowspan="2" class="text-center valign">Username</th>
                    <th  rowspan="2"  class="text-center valign">E-Mail</th>
                    <th  colspan="4" class="text-center valign">Role</th>
                    <th colspan="3"  class="text-center valign">Motorcycle</th>
                    <th rowspan="2" class="text-center valign">Delete</th>
                </tr>
                <tr>
                    <th class="text-center valign">Designation</th>
                    <th  colspan="3" class="text-center valign">Action</th>
                    <th class="text-center valign">Brand</th>
                    <th class="text-center valign">Model</th>
                    <th class="text-center valign">Year</th>
                </tr>


            </thead>
            <tbody id="">
                <?php
                $users = getUsersNMotorcycle();
                foreach ($users as $user) {
                    echo"<tr>";
                    echo "<td>" . $user["Username"] . "</td>";
                    echo "<td>" . $user["email"] . "</td>";
                    $rank = $user["role"];
                    echo "<td > <span id=\"" . $user["idUser"] . "Role\" name=\"$rank\">";

                    switch ($rank) {
                        case 1 : echo "Administrator";
                            break;
                        case 2 : echo "User";
                            break;
                        case 3 : echo "Ban";
                            break;
                    }
                    echo "</span></td>";

                    echo "<td class=\"text-center no-border action-table ";
                    if ($rank == 1) {
                        echo "disabled";
                    }echo "\" style=\"width: 60px;\"><a name=\"" . $user["idUser"] . "\" class=\"UpgradeRole\"><span class=\" glyphicon glyphicon-chevron-up \" > upgrade</span></a> </td>";

                    echo "<td class=\"text-center no-border action-table ";
                    if ($rank != 1) {
                        echo "disabled";
                    }echo "\" style=\"width: 80px;\"><a name=\"" . $user["idUser"] . "\" class=\"DowngradeRole\"><span class=\" glyphicon glyphicon-chevron-down \" > downgrade</span></a></td>";
                    echo "<td class=\"text-center no-border action-table ";
                    if ($rank == 3) {
                        echo "disabled";
                    }echo " \" style=\"width: 50px;\" ><a name=\"" . $user["idUser"] . "\" class=\"BanRole\"><span class=\" glyphicon glyphicon-ban-circle \" > ban</span></a></td>";
                    echo "<td>" . $user["Brand"] . "</td>";
                    echo "<td>" . $user["model"] . "</td>";
                    echo "<td>" . substr($user["year"], 0, 4) . "</td>";
                    echo "<td> <span class=\"glyphicon glyphicon-remove red\" ></span></td>";
                }
                ?>

            </tbody>
            <tfoot>

            </tfoot>

        </table>
          <?php include './Includes/footer.php'; ?>
    </body>
</html>

```

```
<!DOCTYPE html>
<?php
/*
 * Author : Lucien Camuglia
 * Description : managing motorcycle page
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */
?>
<html>
    <head>
        <?php include './Includes/header.php' ?>
        <script src="./JS/ManageMoto.js"></script>
        <title></title>
    </head>
    <body>
        <?php
        DisplayMenu(basename($_SERVER['PHP_SELF']));
        AddMotorcycleModal();
        $motorcyles = GetAllMotorcycles();
        foreach ($motorcyles as $motorcycle) {
            confirmationDeleteMotorcycleModal($motorcycle["idMoto"], $motorcycle["Brand"], $motorcycle["model"], substr($motorcycle["year"], 0, 4));
        }
        ?>

        <table class="table table-striped table-bordered" cellspacing="0" id="MotorcycleTable">
            <thead>
                <tr>
                    <th>Brand <select id="brand" ><option value="%">All</option> <?php
                            $brands = GetMotorcycleBrand();
                            foreach ($brands as $brand) {
                                echo "<option value='" . $brand . "'>" . $brand . "</option>";
                            }
                            ?></select> </th>
                    <th>Model <select  id="model"><option value="%">All</option><?php
                            $models = GetAllMotorcycleModel();
                            foreach ($models as $model) {
                                echo "<option value='" . $model . "'>" . $model . "</option>";
                            }
                            ?></select></th>
                    <th>Year <select  id="year"><option value="%">All</option><?php
                            $years = GetAllMotorcycleYear();
                            foreach ($years as $year) {
                                echo "<option value='" . $year . "'>" . $year . "</option>";
                            }
                            ?></select></th>
                    <th>Consumption (l/100km) <select  id="consumption" ><option value="%">All</option><?php
                            $consumptions = GetAllMotorcycleConsumption();
                            foreach ($consumptions as $consumption) {
                                echo "<option value='" . $consumption . "'>" .
                                $consumption . "</option>";
                            }
                            ?></select></th>
                    <th>Tiredness <select  id="tiredness"><option value="%">All</option><?php
                            $tirednesses = GetAllMotorcycleTiredness();
                            foreach ($tirednesses as $tiredness) {
                                echo "<option value='" . $tiredness . "'>" . $tiredness . "</option>";
                            }
                            ?></select></th>
                    <th>Delete</th>
                </tr>

            </thead>
            <tbody id="motorcylesDatas">
                <?php
                foreach ($motorcyles as $motorcyle) {
                    echo "<tr>";
                    echo "<td>";
                    echo $motorcyle["Brand"];
                    echo "</td>";
                    echo "<td>";
                    echo $motorcyle["model"];
                    echo "</td>";
                    echo "<td>";
                    echo substr($motorcyle["year"], 0, 4);
                    echo "</td>";
                    echo "<td>";
                    echo $motorcyle["consumption"];
                    echo "</td>";
                    echo "<td>";
                    echo $motorcyle["Tiredness"];
                    echo "</td>";
                    echo "<td>";
                    echo "<span class=\"glyphicon glyphicon-remove red\" data-toggle=\"modal\" data-target=\"#modaldelete" . $motorcyle["idMoto"] . "\" ></span>";
                    echo "</td>";
                    echo "</tr>";
                }
                ?>
            </tbody>
            <tfoot>
                <tr><td colspan="6" ><a href="#" data-toggle="modal" data-target="#modalMotorcycle" onclick='$("#msgCont").attr("Hidden", "true")'> + Add a Motorcycle</a></td></tr>
            </tfoot>

        </table>
        <?php include './Includes/footer.php'; ?>
    </body>
</html>

```

```
<!DOCTYPE html>
<?php
/*
 * Author : Lucien Camuglia
 * Description : Signin page
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */
$error = false;
$errorMsg = "";
if (isset($_GET["error"])) {
    if (!empty($_GET["error"])) {
        $error = true;
        $errorMsg = $_GET["error"];
    }
}
$value = false;
if (isset($_GET["values"])) {
    if (!empty($_GET["values"])) {
        $value = true;
        $values = json_decode($_GET["values"]);
    }
}
?>

<html>
    <head>
        <?php include './Includes/header.php' ?>
        <script src="./JS/Signin.js"></script>
        <title></title>
    </head>
    <body style="width: auto;">
        <?php
        DisplayMenu(basename($_SERVER['PHP_SELF']));
        ?>
        <div class="col-sm-6 col-lg-offset-3 ">
            <div class="alert alert-danger" <?php echo ($error) ? "" : "hidden"; ?>>
                <strong>Erreur !</strong> <label id='msg'><?php echo $errorMsg;
                ?></label>
            </div>
            <center>
                <h1>Sign in</h1>
                <form action="./Includes/signIn.php" method="post">
                    <div class="form-group row">
                        <label for="Username" class="col-sm-3 col-form-label"
                        >Username  <span data-toggle="tooltip" title="min 5 char"
                        class="glyphicon glyphicon-info-sign"
                        id="InfoUsername"></span> </label>
                        <div class="col-sm-8">
                            <input class="form-control" type="text" name="username"
                            id="Username" value="<?php
                            if ($value) {
                                if (isset($values->username)) {
                                    echo $values->username;
                                }
                            }
                            ?>">
                            </div>

                            <span class="glyphicon glyphicon-remove red" id="LogoUsername"></span>
                        </div>
                        <div class="form-group row">
                            <label for="Username" class="col-sm-3 col-form-label">email</label>
                            <div class="col-sm-8">
                                <input class="form-control" type="email" name="email" id="email" value="<?php
                                if ($value) {
                                    if (isset($values->email)) {
                                        echo $values->email;
                                    }
                                }
                                ?>">
                            </div>
                            <span class="glyphicon glyphicon-remove red" id="LogoEmail"></span>
                        </div>
                        <div class="form-group row">
                            <label for="Username" class="col-sm-3 col-form-label">password <span data-toggle="tooltip" title="8char min, 1upper, 1lower, 1Numeric" class="glyphicon glyphicon-info-sign" id="InfoPwd"></span> </label>
                            <div class="col-sm-8">
                                <input class="form-control" type="password" value="" name="password" id="password">
                            </div>

                            <span class="glyphicon glyphicon-remove red" id="Logopwd"></span>
                        </div>
                        <div class="form-group row">
                            <label for="Username" class="col-sm-3 col-form-label">password confirmation</label>
                            <div class="col-sm-8">
                                <input class="form-control" type="password" value="" name="password2" id="password2">
                            </div>
                            <span class="glyphicon glyphicon-remove red" id="LogoPwd2"></span>
                        </div>

                        <div class="panel panel-primary">
                            <div class="panel-heading">Motorcycle</div>
                            <div class="panel-body">
                                <div class="form-group row">
                                    <label for="Username" class="col-sm-3 col-form-label ">Brand</label>
                                    <div class="col-sm-9">
                                        <select class='brand' name="brand">
                                            <?php
                                            if ($value) {
                                                if (isset($values->brand)) {
                                                    echo "<option
                            value=\"$values->brand\" >$values->brand</option>";
                        }
                    }
                    ?>
                    <option value="">Select...</option>
                    <?php
                    $brands = GetMotorcycleBrand();
                    foreach ($brands as $brand) {
                        echo "<option value='" . $brand . "'>" . $brand . "</option>";
                    }
                    ?>
                </select>
            </div>
        </div>

        <div class="form-group row">
            <label for="Username" class="col-sm-3 col-form-label">Model </label>
            <div class="col-sm-9">
                <select id="model" name="model">
                    <?php
                    if ($value) {
                        if (isset($values->model)) {
                            echo "<option value=\"$values->model\" >$values->model</option>";
                        }
                    }
                    ?>

                    <option  value="">Select a brand</option>
                </select>
            </div>
        </div>

        <div class="form-group row">
            <label for="Username" class="col-sm-3 col-form-label">Year </label>
            <div class="col-sm-9">
                <select id="year" name="year" >
                    <?php
                    if ($value) {
                        if (isset($values->year)) {
                            echo "<option value=\"$values->year\" >$values->year</option>";
                        }
                    }
                    ?>
                    <option  value="">Select a model</option>
                </select>
            </div>
        </div>
    </div>
</div>
```

```
                    <div class="form-group row">
                        <div class="col-sm-4 col-lg-offset-4">
                            <input class="form-control btn btn-primary disabled " type="submit" name="submit" value="Sign in" id="Signin">
                        </div>
                    </div>
                </form>
            </center>
        </div>
        <?php include './Includes/footer.php'; ?>
    </body>
</html>

```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : log out an user
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

session_start();
session_destroy();
header("location: ./index.php");


```

```javascript
/*
 * Author : Lucien Camuglia
 * Description : All map fonctions
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */


//Google API KEYS, DEFAULT is my default key for this project ans RESCUE is used only if DEFAULT'S quotas has been reached
var KEY = (function() {
    var private = {
        'DEFAULT': 'AIzaSyCRxYbU0CGNMpZINbtBJqn72k1UCi0bMo8',
        'RESCUE': 'AIzaSyAe7EU4muOht_cbrvH88JTd2FTrvbsYZ_E'
    };
    return {
        get: function(name) {
            return private[name];
        }
    };
})();


var map;
var mapkey = KEY.get('DEFAULT');
var polyline;
var pointsArray = [];
var midmarkers = [];
var oldMarkers = [];
var Markers = [];
var oldRoute = [];
var route = [];
var highlighted;
var addedPoints = 0;
var directionsDisplay = new google.maps.DirectionsRenderer({suppressMarkers: true});
var directionsService = new google.maps.DirectionsService();
var elevator = new google.maps.ElevationService;
var parcoursPolyline;
var imageNormal = new google.maps.MarkerImage(
        "./images/lines/square.png",
        new google.maps.Size(11, 11),
        new google.maps.Point(0, 0),
        new google.maps.Point(6, 6)
        );
var imageHover = new google.maps.MarkerImage(
        "./images/lines/square_over.png",
        new google.maps.Size(11, 11),
        new google.maps.Point(0, 0),
        new google.maps.Point(6, 6)
        );
var imageNormalMidpoint = new google.maps.MarkerImage(
        "./images/lines/square_transparent.png",
        new google.maps.Size(11, 11),
        new google.maps.Point(0, 0),
        new google.maps.Point(6, 6)
        );
var total = 0;
var DisplayTraffic = false;
var avoidHighways = false;

//initialise the map
function initMap(modif, traffic) {
    modif = modif || false;
    traffic = traffic || false;

    var Searchinput = document.getElementById('pac-input');
    var searchBox = new google.maps.places.SearchBox(Searchinput);

    var pos = {
        lat: 46.203545,
        lng: 6.145150
                //geneva
    };

    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(function(position) {
            pos = {
                lat: position.coords.latitude,
                lng: position.coords.longitude
            };
        })
    }

    map = new google.maps.Map(document.getElementById('map'), {
        zoom: 4,
        center: pos,
        mapTypeControl: true,
        mapTypeControlOptions: {
            style: google.maps.MapTypeControlStyle.DROPDOWN_MENU,
            mapTypeId: google.maps.MapTypeId.TERRAIN
        }
    });
    //if the route can be modified
    if (modif) {
        google.maps.event.addListener(map, 'click', function(event) {
            CreateMarker(event.latLng, true, true, "FE7569", ++addedPoints, addedPoints);
            $("#pointsInfo").append("<div id=point" + addedPoints + ">\n\
                                     <div class=\"Pointname col-lg-12\" >\n\
                                     <span class=\"col-lg-2 align-middle\" style=\"height: 100%;\" >Point  " + addedPoints + ": </span>\n\
                                     <div class=\"col-lg-10\">\n\
                                    <div> <label class=\"col-sm-3\">Latitude : </label> <input id='inpLatPoint" + addedPoints + "' type=\"text\" value=" + event.latLng.lat() + "> </div>\n\
                                    <div> <label class=\"col-sm-3\">Longitude : </label> <input id='inpLonPoint" + addedPoints + "' type=\"text\" value=" + event.latLng.lng() + ">\n\
                                       </div></div><div></div>   ");
            if (addedPoints > 1) {
                searchIti(Markers[addedPoints - 2].position, Markers[addedPoints -
                1].position);
            }
        });
        Searchinput.removeAttribute('hidden');
        map.controls[google.maps.ControlPosition.TOP_LEFT].push(Searchinput);
        map.addListener('bounds_changed', function() {
            searchBox.setBounds(map.getBounds());
        });
        searchBox.addListener('places_changed', function() {
            var places = searchBox.getPlaces();
            if (places.length == 0) {
                return;
            }
            var bounds = new google.maps.LatLngBounds();
            places.forEach(function(place) {
                if (!place.geometry) {
                    return;
                }
                var icon = {
                    url: place.icon,
                    size: new google.maps.Size(71, 71),
                    origin: new google.maps.Point(0, 0),
                    anchor: new google.maps.Point(17, 34),
                    scaledSize: new google.maps.Size(25, 25)
                };
                if (place.geometry.viewport) {
                    // Only geocodes have viewport.
                    bounds.union(place.geometry.viewport);
                } else {
                    bounds.extend(place.geometry.location);
                }

            });
            map.fitBounds(bounds);
        });

    }
    //if the traffic have to be diplay
    if (traffic) {
        var trafficLayer = new google.maps.TrafficLayer();
        trafficLayer.setMap(map);
    }
}

//import a GPX file
function ImporterGpx(file) {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "ImportGPX", file: file},
        dataType: "json",
        success: function(idRoute) {

        }
    });
}

```

```
//Ask google Roads for snapping point to the road
function AskGoogle(Path) {
    var snapped = [];
    $.ajax({
        url: 'https://roads.googleapis.com/v1/snapToRoads',
        type: 'GET',
        data: {path: Path, interpolate: true, key: mapkey},
        dataType: "json",
        async: false,
        success: function(r2) {

            $.each(r2["snappedPoints"], function(index, value)
            {
                snapped[index] = value.location;
            });
        }
    });
    return snapped;
}

//load all points of a route
function LoadPoints(idRoute) {
    var resultArray = [];
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "GetRoutePoints", idRoute: idRoute},
        dataType: "json",
        async: false,
        success: function(result) {
            $.each(result, function(index, value)
            {
                resultArray[index] = value;
            });
        }
    });
    return resultArray;
}

//snapp all points of the route to the road
function SnappPoints2Road(route) {
    var Path = "";
    var tmpSnapped = [];
    var Snapped = [];
    $.each(route, function(index, point)
    {
        if ((index > 0) && (index % 50 === 0)) {
            Path = Path.slice(0, -1);
            tmpSnapped = AskGoogle(Path);
            Path = "";
            Snapped = Snapped.concat(tmpSnapped);
        }
        Path += point.Latitude + "," + point.Longitude + "|";
    });
    Path = Path.slice(0, -1);
    Snapped = Snapped.concat(AskGoogle(Path));
    return Snapped;
```

```
}

//save a new location
function SaveNewLocation(idroute, route, fromweb) {
    fromweb = fromweb || false;
    var geoRoute = [];
    var echantRoute = [];
    var tmpRoute = route;
    $.each(route, function(index, value)
    {
        //verify if the request is made from the web api
        if (fromweb) {
            geoRoute.push(new google.maps.LatLng(value.lat(), value.lng()));

        } else {
            geoRoute.push(new google.maps.LatLng(value.latitude, value.longitude));

        }
    });
    //get length and calcul the sinuosity
    var length = google.maps.geometry.spherical.computeLength(geoRoute);
    var sinuosity = route.length / length;
    var counter = 10;

    if (geoRoute.length <= 10) {
        counter = 1;
    }
    if (geoRoute.length > 4000) {
        counter = 20;
    }
    if (geoRoute.length > 8000) {
        counter = 30;
    }
    if (geoRoute.length > 12000) {
        counter = 50;
    }

    for (var i = 0; i < geoRoute.length - counter; i += counter) {
        echantRoute.push(geoRoute[i]);
    }
    //get the levation of each point of the route
    elevator.getElevationAlongPath({
        'path': echantRoute,
        'samples': 256
    }, function(elevations, status) {
        if (status !== 'OK') {
            console.log("Error elevaton : " + status);
        } else {
            var elevationAverage = 0;
            //calcul the elevation average
            for (var i = 1; i < elevations.length; i++) {
                elevationAverage += Math.abs(elevations[i - 1].elevation -
                elevations[i].elevation);
            }
            if (fromweb) {
                route = JSON.stringify(route);
            }
            //send the infos to the PHP file
            $.ajax({
                url: './Includes/ajax.php',
                type: 'POST',
                data: {fonction: "SaveNewRoute", idRoute: idroute, route: route, sinuosite: sinuosity, elevation: elevationAverage, length: length},
                async: false,
                success: function(result) {
                    if (fromweb) {
                        location.reload();
                    }
                }
            });
        }
    });
}

//call the PHP fonction for creating a new route
function CreateRoute(name, containsHighway) {
    if (name != "") {
        $.ajax({
            url: './Includes/ajax.php',
            type: 'GET',
            data: {fonction: "CreateRoute", name: name, containsHighway: containsHighway},
            dataType: 'json',
            async: false,
            success: function(result) {
                SaveNewLocation(result, route, true);
            }
        });
    }
}

$(document).ready(function() {
    initMap();
    clear();
    RouteClick();
    //when the modification button is pressed
    $("#btnModif").click(function() {
        if (highlighted != null) {
            oldRoute = route;
            oldMarkers = Markers;
            clear();
            initMap(true);
            ShowParcours(LoadPoints($(this).attr('name')), true);
            $(".modifRoute").removeClass("hidden");
            $("#inpLatPoint0").val(Markers[0].position.lat());
            $("#inpLonPoint0").val(Markers[0].position.lng());
            $("#inpLatPoint1").val(Markers[1].position.lat());
            $("#inpLonPoint1").val(Markers[1].position.lng());
            editlines();
        }
    });
    //when th refresh button is pressed
    $("#RefreshRoute").click(function() {
        if (oldMarkers.length === Markers.length) {
            var j = 0;
            for (var i = 0; i < Markers.length; i++) {
                var from = [];
                from["lat"] = Markers[i].position.lat();
                from["lon"] = Markers[i].position.lng();
                var to = [];
                to["lat"] = oldMarkers[i].position.lat();
                to["lon"] = oldMarkers[i].position.lng();
                searchIti(from, to);
                if (i < Markers.length - 1)
                    while ((oldRoute[j].lat() !== oldMarkers[i + 1].position.lat()) && (oldRoute[j].lng() !== oldMarkers[i + 1].position.lng())) {
                        route.push(oldRoute[j]);
                        j++;
                    }
                else
                    while (j < oldRoute.length) {
                        route.push(oldRoute[j]);
                        j++;
                    }
            }

        }
    });

    $("#sinuosity").change(function() {
        RefreshhRoutesWithFilters();
    });
    $("#slope").change(function() {
        RefreshhRoutesWithFilters();
    });
    $("#highway").change(function() {
        RefreshhRoutesWithFilters();
    });
    $("#StartCreation").click(function() {
        StartCreation();

    });
    $("#ClearRoute").click(function() {
        ClearRoute();
    });

    $("#chkHighway").change(function() {
        if ($("#chkHighway").is(':checked')) {
            avoidHighways = false;
        } else {
            avoidHighways = true;
        }
    });
});

//when a request AJAX start, display a wheel
$(document).bind({
    ajaxStart: function() {
        $("body").addClass("loading");
    },
    ajaxStop: function() {
        $("body").removeClass("loading");
    }
});

function RouteClick() {
    $(".route").click(function() {
        clear();
        initMap(false, DisplayTraffic);
        $("#DeleteRoute").click(function() {
            DeleteRoute($(highlighted).attr('name'));

        });
        $("#btnModif").removeClass("hidden");
        $(".routeControl").removeClass("hidden");
        $("#btnModif").attr('name', $(this).attr('name'));
        if (highlighted != null)
            $(highlighted).removeClass("highlight");
        $(this).addClass("highlight");
        highlighted = this;
        ShowParcours(LoadPoints($(this).attr('name')), false);
        DisplayRouteInfo($(this).attr('name'));

    });
}

//search an itinary
function searchIti(from, to) {

    var request = {
        origin: from,
        destination: to,
        travelMode: google.maps.TravelMode.DRIVING,
        avoidHighways: avoidHighways
                //WALKING / DRIVING / BICYCLING / TRANSIT /
    };

    directionsDisplay.setMap(this.map);
    directionsService.route(request, function(result, status) {

        if (status === google.maps.DirectionsStatus.OK) {
            var distanceM = result.routes[0].legs[0].distance.value;
            var tempsS = result.routes[0].legs[0].duration.value;
            //add all point in the current route
            $.each(result.routes[0].overview_path, function(index, point) {
                route.push(point);
            });
            //display the route
            DisplayRoute();
        } else {
            console.log(status);
        }

    });
}

/* display the points
 @param array tableauPoints : Markers array to display
 @param bool parcoursModif : if the road can be modified
 */
function ShowParcours(tableauPoints, parcoursModif, color) {
    color = color || "#ff1ece";
    var liste_des_points = tableauPoints;
    if (parcoursModif === true) {
        //create the first and last markers
        CreateMarker(liste_des_points[0], parcoursModif, false, "36af2d", "%E2%80%A2", 0);
        CreateMarker(liste_des_points[liste_des_points.length - 1], parcoursModif, false, "a52424", "%E2%80%A2", 1);
    }

    $.each(liste_des_points, function(index, value)
    {
        route.push(new google.maps.LatLng(value.Latitude, value.Longitude));
    });
    var bounds = new google.maps.LatLngBounds();
    for (var i = 0; i < liste_des_points.length; i++) {
        bounds.extend(new google.maps.LatLng(liste_des_points[i].Latitude, liste_des_points[i].Longitude));
    }
    map.fitBounds(bounds);
    DisplayRoute(color);
}

//Display the polyline
function DisplayRoute(color) {
    color = color || "#ff1ece";
    parcoursPolyline = new google.maps.Polyline({
        path: route,
        strokeColor: color,
        strokeOpacity: 1.0,
        strokeWeight: 5,
        map: map
    });
}

//create a new marker
function CreateMarker(value, parcoursModif, parcoursDelete, pinColor, texte, position) {
    //load the image and add the position in the road
    var pinImage = new google.maps.MarkerImage("http://chart.apis.google.com/chart?chst=d_map_pin_letter&chld=" + texte + "|" + pinColor,
            new google.maps.Size(21, 34),
            new google.maps.Point(0, 0),
            new google.maps.Point(10, 34));
    var pos;
    if (value.Latitude) {
        pos = new google.maps.LatLng(value.Latitude, value.Longitude);
    } else {
        pos = value;
    }

    parcoursMarker = new google.maps.Marker({
        position: pos,
        draggable: parcoursModif,
        zIndex: position,
        map: map,
        icon: pinImage
    });
    //if the marker can be delete
    if (parcoursDelete) {
        parcoursMarker.addListener('click', function() {
            this.setMap(null);
            Markers.splice(Markers.indexOf(this), 1);
            $("#point" + this.zIndex).remove();
            length--;
            addedPoints--;
        });
    }

    // if the marker can be move
    if (parcoursModif) {
        parcoursMarker.addListener('dragend', function() {
            refreshValues(this.zIndex);
            Markers[this.zIndex].previous = Markers[this.zIndex].position;
            Markers[this.zIndex].latLng = this.position;
        });
    }

    //add the marker in the table
    Markers.push(parcoursMarker);
}

//display the values in the fields
function refreshValues(index) {
    $("#inpLatPoint" + index).val(Markers[index].position.lat());
    $("#inpLonPoint" + index).val(Markers[index].position.lng());
}

//Download the route
function DownloadRoute() {
    if (highlighted == null) {
        alert("Please select a trip");
    } else
    {
        name = $(highlighted).html();
        name = name.substr(0, name.indexOf("<span"));
        $.ajax({
            url: './Includes/ajax.php',
            type: 'POST',
            data: {fonction: "Download", name: name, path: JSON.stringify(route)},
            dataType: "html",
            async: false,
            success: function(result) {
                document.location.href = "./Includes/download.php?file=" + result;
            },
            error: function(result, status, error) {
                console.log(result);
                console.log(status);
                console.log(error);
            }
        });
    }
}

//display or not the filters
function EnabledDisabledFilters(caller) {
    if (caller.checked) {
        $("#filters").removeAttr("hidden");
        RefreshhRoutesWithFilters();
    } else {
        $("#filters").attr("hidden", true);
        RefreshhRoutesWithoutFilters();
    }
}

//refresh the displayed routes depending to the filters
function RefreshhRoutesWithFilters() {
    var sinuosity = $("#sinuosity").val();
    var slope = $("#slope").val();
    var highway = $("#highway").prop("checked");
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "FilterRoad", sinuosity: sinuosity, slope: slope, highway: highway, time: 1},
        dataType: "json",
        success: function(result) {
            $(".routes").empty();
            $.each(result, function(index, value)
            {
                $(".routes").append("<div class=\"route\" name=\"" + value.idRoute + "\">" + value.RouteName + "<span class=\"by\"> by " + value.Username + "</span></div>");
            });
            RouteClick();
        }

    });
}

//refresh the displayed routes
function RefreshhRoutesWithoutFilters() {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "GetRoutes"},
        dataType: "json",
        async: false,
        success: function(result) {
            $(".routes").empty();
            $.each(result, function(index, value)
            {
                $(".routes").append("<div class=\"route\" name=\"" + value.idRoute + "\">" + value.RouteName + "<span class=\"by\"> by " + value.Username + "</span></div>");
            });
            RouteClick();
        },
        error: function(XMLHttpRequest, textStatus, errorThrown) {
            alert("Status: " + textStatus);
            alert("Error: " + errorThrown);
        }
    });
}

//start the creation of the new route (display the buttons
function StartCreation() {
    ClearRoute();
    $(".SaveRouteControls").removeClass('hidden');
}

//display all infos about the road
function DisplayRouteInfo(idroute) {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "GetRoadsInfos", idRoute: idroute},
        dataType: "json",
        async: false,
        success: function(result) {
            $('#InfoHighway').html((result.Highway == 1) ? 'This road includes highways' : 'This road does not include highways');
            $('#InfoLength').html("Length : " + result.Length + "km");
            $('#InfoSinuosity').html("Sinuosity : " + result.Sinuosity + "/10");
            $('#InfoSlope').html("Slope : " + result.Slope + "/10");
            $('#InfoDuration').html("Travel time : " + result.Time);
            if (result.MotorcycleConsumption == 0) {
                $('#InfoConsumption').html("Consumption : Please login for more informations");
            } else {
                $('#InfoConsumption').html("Théorical consumption : " + Math.round(result.Length * result.MotorcycleConsumption / 100) + " liters");
            }

        },
        error: function(result, status, error) {
            console.log(result);
            console.log(status);
            console.log(error);
        }
    });

}

//call the PHP fonction for delete a route
function DeleteRoute(idRoute) {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "DeleteRoute", idRoute: idRoute},
        dataType: 'json',
        async: false,
        success: function(result) {
                location.reload();
            }
        });
    }

    //clear all fields
    function ClearRoute() {
        initMap(true, DisplayTraffic);
        pointsArray = [];
        midmarkers = [];
        oldMarkers = [];
        oldRoute = [];
        addedPoints = 0;
        clear();
    }

    function clear() {
        Markers = [];
        route = [];
    }





    // Editting the lines, found on :
    http://www.birdtheme.org/useful/v3tool.html
    function editlines() {
        Markers = [];
        route = parcoursPolyline.getPath();
        if (route.length > 0) {
            for (var i = 0; i < route.length; i++) {
                var marker = setmarkers(route.getAt(i));
                Markers.push(marker);
                if (i > 0) {
                    var midmarker = setmidmarkers(route.getAt(i));
                    midmarkers.push(midmarker);
                }
            }
        }
    }
    function setmarkers(point) {
        var marker = new google.maps.Marker({
            position: point,
            map: map,
            icon: imageNormal,
            raiseOnDrag: false,
            draggable: true
        });
        google.maps.event.addListener(marker, "mouseover", function() {
            marker.setIcon(imageHover);
        });
        google.maps.event.addListener(marker, "mouseout", function() {
            marker.setIcon(imageNormal);
        });
        google.maps.event.addListener(marker, "drag", function() {
            for (var i = 0; i < Markers.length; i++) {
                if (Markers[i] == marker) {
                    parcoursPolyline.getPath().setAt(i, marker.getPosition());
                    movemidmarker(i);
                    break;
                }
            }
            route = parcoursPolyline.getPath();
            var stringtobesaved = marker.getPosition().lat().toFixed(6) + ',' + marker.getPosition().lng().toFixed(6);
            pointsArray.splice(i, 1, stringtobesaved);
        });
        google.maps.event.addListener(marker, "click", function() {
            for (var i = 0; i < Markers.length; i++) {
                if (Markers[i] == marker && Markers.length != 1) {
                    marker.setMap(null);
                    Markers.splice(i, 1);
                    parcoursPolyline.getPath().removeAt(i);
                    //    removemidmarker(i);
                    break;
                }
            }
            polyPoints = parcoursPolyline.getPath();
            if (Markers.length > 0) {
                pointsArray.splice(i, 1);
                logCode1();
            }
        });
        return marker;
    }
    function setmidmarkers(point) {
        var prevpoint = Markers[Markers.length - 2].getPosition();
        var marker = new google.maps.Marker({
            position: new google.maps.LatLng(
                    point.lat() - (0.5 * (point.lat() - prevpoint.lat())),
                    point.lng() - (0.5 * (point.lng() - prevpoint.lng()))
                    ),
            map: map,
            icon: imageNormalMidpoint,
            raiseOnDrag: false,
            draggable: true
        });
        google.maps.event.addListener(marker, "mouseover", function() {
            marker.setIcon(imageNormal);
        });
        google.maps.event.addListener(marker, "mouseout", function() {
            marker.setIcon(imageNormalMidpoint);
        });
        google.maps.event.addListener(marker, "dragend", function() {
            for (var i = 0; i < midmarkers.length; i++) {
                if (midmarkers[i] == marker) {
                    var newpos = marker.getPosition();
                    var startMarkerPos = Markers[i].getPosition();
                    var firstVPos = new google.maps.LatLng(
                            newpos.lat() - (0.5 * (newpos.lat() - startMarkerPos.lat())),
                            newpos.lng() - (0.5 * (newpos.lng() - startMarkerPos.lng()))
                            );
                    var endMarkerPos = Markers[i + 1].getPosition();
                    var secondVPos = new google.maps.LatLng(
                            newpos.lat() - (0.5 * (newpos.lat() - endMarkerPos.lat())),
                            newpos.lng() - (0.5 * (newpos.lng() - endMarkerPos.lng()))
                            );
                    var newVMarker = setmidmarkers(secondVPos);
                    newVMarker.setPosition(secondVPos); //apply the correct position to the midmarker
                    var newMarker = setmarkers(newpos);
                    Markers.splice(i + 1, 0, newMarker);
                    parcoursPolyline.getPath().insertAt(i + 1, newpos);
                    marker.setPosition(firstVPos);
                    midmarkers.splice(i + 1, 0, newVMarker);
                    break;
                }
            }
            polyPoints = parcoursPolyline.getPath();
            var stringtobesaved = newpos.lat().toFixed(6) + ',' + newpos.lng().toFixed(6);
            pointsArray.splice(i + 1, 0, stringtobesaved);
            logCode1();
        });
        return marker;
    }
    function movemidmarker(index) {
        var newpos = Markers[index].getPosition();
        if (index != 0) {
            var prevpos = Markers[index - 1].getPosition();
            midmarkers[index - 1].setPosition(new google.maps.LatLng(
                    newpos.lat() - (0.5 * (newpos.lat() - prevpos.lat())),
                    newpos.lng() - (0.5 * (newpos.lng() - prevpos.lng()))
                    ));
        }
        if (index != Markers.length - 1) {
            var nextpos = Markers[index + 1].getPosition();
            midmarkers[index].setPosition(new google.maps.LatLng(
                    newpos.lat() - (0.5 * (newpos.lat() - nextpos.lat())),
                    newpos.lng() - (0.5 * (newpos.lng() - nextpos.lng()))
                    ));
        }
    }
    function removemidmarker(index) {
        if (Markers.length > 0) {//clicked marker has already been deleted
            if (index != Markers.length) {
                midmarkers[index].setMap(null);
                midmarkers.splice(index, 1);
            } else {
                midmarkers[index - 1].setMap(null);
                midmarkers.splice(index - 1, 1);
            }
        }
        if (index != 0 && index != Markers.length) {
            var prevpos = Markers[index - 1].getPosition();
            var newpos = Markers[index].getPosition();
            midmarkers[index - 1].setPosition(new google.maps.LatLng(
                    newpos.lat() - (0.5 * (newpos.lat() - prevpos.lat())),
                    newpos.lng() - (0.5 * (newpos.lng() - prevpos.lng()))
                    ));
        }
```

```
}


```

```
/*
 * Author : Lucien Camuglia
 * Description : Sign in javascript fonctions
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

//field validation regex
var passwordRegex = new RegExp(/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8,}$/i); //8char 1upper 1 lower 1 number
var emailRegex = new RegExp(/^([a-zA-Z0-9_.+-])+\@(([a-zA-Z0-9-])+\.)+([a-zA-Z0-9]{2,4})+$/i); //found on http://stackoverflow.com/questions/2507030/email-validation-using-jquery

/**
 * Connexion function
 * @param {array} datas array who contains username and data
 */
function connexion(datas) {
    $.ajax({
        url: './Includes/connexion.php',
        type: 'POST',
        data: {username: datas[0], password: datas[1]},
        dataType: "json",
        success: function(result) {
            if (result.status === "success") {
                location.reload();
            } else {
                $("#msgCont").removeAttr("hidden");
                $("#msg").html(result.message);
            }
        }
    });
}

//Load motorcycle model from brand
function LoadModelFromBrand(brand) {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "GetModel", Brand: brand},
        dataType: "json",
        success: function(result) {
            //clear the select
            $("#model").html("");
            $("#year").html("<option value=\"\">Select a model</option>");
            $("#model").append("<option value=\"\">Select...</option>");
            //add all returned motorcycle
            $.each(result, function(i, item) {
                $("#model").append("<option value=" + item + ">" + item + "</option>");
            });
        }
    });
}
//Load motorcycle year from brand and model
function LoadYearFromModel(brand, model) {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "GetYear", Brand: brand, Model: model},
        dataType: "json",
        success: function(result) {
            $("#year").html("");
            $("#year").append("<option value=\"\">Select...</option>");
            $.each(result, function(i, item) {
                $("#year").append("<option value=" + item + ">" + item + "</option>");
            });
        }
    });


}

//check if the username exists
function UserNameExists(username) {
    exist = false;
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "UserExists", Username: username},
        dataType: "json",
        async: false,
        timeout: 30000,
        success: function(result) {
            if (result.exist) {
                exist = true;
            }
        }
    });
    return exist;
}

//check if the submit button can be pressed
function checkForEnabled() {
    $("#Signin").removeClass("disabled");
    if ($("#LogoUsername").hasClass("red")) {
        $("#Signin").addClass("disabled");
    }
    if ($("#Logopwd").hasClass("red")) {
        $("#Signin").addClass("disabled");
    }
    if ($("#LogoEmail").hasClass("red")) {
        $("#Signin").addClass("disabled");
    }
    if ($("#LogoPwd2").hasClass("red")) {
        $("#Signin").addClass("disabled");
    }
    if ($(".brand").val() === "") {
        $("#Signin").addClass("disabled");
    }
    if ($("#model").val() === "") {
        $("#Signin").addClass("disabled");
    }
    if ($("#year").val() === "") {
        $("#Signin").addClass("disabled");
    }
}

$(document).ready(function() {

    //active Tooltip
    $('[data-toggle="tooltip"]').tooltip();

    $(".brand").click(function() {
        LoadModelFromBrand($(this).val());
        checkForEnabled();
    });

    $("#model").click(function() {
        LoadYearFromModel($(".brand").val(), $(this).val());
        checkForEnabled();
    });

    $("#year").click(function() {
        checkForEnabled();
    });

    //check if the username exists or not when the input change
    $('#Username').on('input', function(e) {
        text = $("#Username").val();
        $("#LogoUsername").removeClass("green");
        $("#LogoUsername").removeClass("glyphicon-ok");
        $("#LogoUsername").addClass("red");
        $("#LogoUsername").addClass("glyphicon-remove");
        if (text.length >= 5) {
            exist = UserNameExists(text);
            console.log(exist);
            if (!exist) {
                $("#LogoUsername").removeClass("red");
                $("#LogoUsername").removeClass("glyphicon-remove");
                $("#LogoUsername").addClass("green");
                $("#LogoUsername").addClass("glyphicon-ok");
            }
        }
        checkForEnabled();
    });

    //check if the password is correct or not when the input change
    $('#password').on('input', function(e) {
        $("#Logopwd").removeClass("green");
        $("#Logopwd").removeClass("glyphicon-ok");
        $("#Logopwd").addClass("red");
        $("#Logopwd").addClass("glyphicon-remove");
        if (passwordRegex.test($('#password').val())) {
            $("#Logopwd").removeClass("red");
            $("#Logopwd").removeClass("glyphicon-remove");
            $("#Logopwd").addClass("green");
            $("#Logopwd").addClass("glyphicon-ok");
        }
        checkForEnabled();
    });

   //check if the email is correct or not when the input change
    $('#email').on('input', function(e) {
        $("#LogoEmail").removeClass("green");
        $("#LogoEmail").removeClass("glyphicon-ok");
        $("#LogoEmail").addClass("red");
        $("#LogoEmail").addClass("glyphicon-remove");
        if (emailRegex.test($('#email').val())) {
            $("#LogoEmail").removeClass("red");
            $("#LogoEmail").removeClass("glyphicon-remove");
            $("#LogoEmail").addClass("green");
            $("#LogoEmail").addClass("glyphicon-ok");
        }
        checkForEnabled();
    });

    //check if the 2nd password equals the 1st or not when the input change
    $('#password2').on('input', function(e) {
        $("#LogoPwd2").removeClass("green");
        $("#LogoPwd2").removeClass("glyphicon-ok");
        $("#LogoPwd2").addClass("red");
        $("#LogoPwd2").addClass("glyphicon-remove");
        if ($('#password2').val() === $('#password').val()) {
            $("#LogoPwd2").removeClass("red");
            $("#LogoPwd2").removeClass("glyphicon-remove");
            $("#LogoPwd2").addClass("green");
            $("#LogoPwd2").addClass("glyphicon-ok");
        }
        checkForEnabled();
    });

});
```

```
/*
 * Author : Lucien Camuglia
 * Description : managing users fonctions
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

//update the user role
function ModifyUser(idUser, role) {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "UpdateUserRole", idUser: idUser, Role: role},
        dataType: "json",
        success: function(result) {
            $("#" + idUser + "Role").html(result.datas.NewRole);
            location.reload();
        }
    });
}


$(document).ready(function() {
    $(".BanRole").click(function() {
        var idUser = $(this).attr('name');
        ModifyUser(idUser, 3);
    });

    $(".UpgradeRole").click(function() {
        var idUser = $(this).attr('name');
        var role = $("#" + idUser + "Role").attr('name');
        ModifyUser(idUser, --role);
    });

    $(".DowngradeRole").click(function() {
        var idUser = $(this).attr('name');
        var role = $("#" + idUser + "Role").attr('name');
        ModifyUser(idUser,++role);
    });
});
```

```
/*
 * Author : Lucien Camuglia
 * Description : managing motorcycles fonctions
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

//create a new motorcycle
function AddMotorcycle(brand, model, year, consumption, tiredness) {

    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "AddMotorcycle", Brand: brand, Model: model, Year: year, Consumption: consumption, Tiredness: tiredness},
        dataType: "json",
        async: false,
        success: function(result) {
            if (result.error.status === true) {
            } else {
                window.location.reload(false);
            }

        }
    });
}


//load all motorcycles
function LoadMotorcycles(brand, model, year, consumption, tiredness) {
    $.ajax({
        url: './Includes/ajax.php',
        type: 'GET',
        data: {fonction: "GetMotorcycles", Brand: brand, Model: model, Year: year, Consumption: consumption, Tiredness: tiredness},
        dataType: "json",
        async: false,
        success: function(result) {
            $("#motorcylesDatas").html("");
            $.each(result, function(i, item) {

                $("#motorcylesDatas").append("<tr> <td>" + item.Brand + "</td>\n\
                                                 <td>" + item.model + "</td>\n\
                                                 <td>" + item.year.substring(0, 4) + "</td> \n\
                                                <td>" + item.consumption + "</td>\n\
                                                 <td>" + item.Tiredness + "</td>\n\
                                                <td> <span class=\"glyphicon glyphicon-remove red\" ></span> </td></tr>");

            });
        }
    });
}


```

```
$(document).ready(function() {
    //filters for motorcycles
    $("select").click(function() {
        LoadMotorcycles($("#brand").val(), $("#model").val(), $("#year").val() +
        "-01-01", $("#consumption").val(), $("#tiredness").val());
    });
});

```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : header of html pages
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

/*
 * ******************************************************
                           CLEF API
 * ******************************************************
  AIzaSyCRxYbU0CGNMpZINbtBJqn72k1UCi0bMo8
 */
session_start();
include './Includes/menu.php';
include './Includes/functions.php';
?>
<meta charset="UTF-8">
<meta name="description" content="Sharng road between motorcycle enthousiasts">
<meta name="keywords" content="Motorcycle,road,moto,ride,findyourride">
<meta name="author" content="Lucien Camuglia">

<link rel="stylesheet" href="./Includes/bootstrap/css/bootstrap.min.css"/>
<link rel="stylesheet" href="./CSS/main.css"/>
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCRxYbU0CGNMpZINbtBJqn72k1UCi0bMo8&libraries=geometry,places&v=3&sensor=false"></script>
<script src="./JS/jquery.min.js"></script>
<script src="./JS/Signin.js"></script>
<link href="./Includes/bootstrap/bootstrap-toggle/css/bootstrap-toggle.min.css" rel="stylesheet">
<link rel="icon" type="image/png" href="./images/logo.png" />

<script src="./Includes/bootstrap/bootstrap-toggle/js/bootstrap-toggle.min.js"></script>
<script src="https://github.com/douglascrockford/JSON-js/blob/master/json2.js"></script>


<script src="./Includes/bootstrap/js/bootstrap.min.js"></script>

```

```
<footer class="footer">
    <span>Lucien Camuglia &copy; CFPT Technician Diploma 2017</span><span style="text-align: right;"><a href="https://github.com/LucienCamuglia/FindYourRide" target = "_blank">Follow me on GitHub</a></span>
</footer>

```

```
<?php
/*
 * Author : Lucien Camuglia
 * Description : navigation menu
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */
require_once './Includes/Modals.php';

function DisplayMenu($page) {
    includeConnexionModal();
    $erreur = false;
    if (isset($_SESSION['erreur'])) {
        $erreur = true;
        $erreurMsg = $_SESSION['erreur'];
        session_unset($_SESSION['erreur']);
    }
    ?>
    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed"
                data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"
                aria-expanded="false">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="./Index.php">Find your ride</a>
            </div>


            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                <ul class="nav navbar-nav">
                    <li <?php
                    if ($page == "Index.php") {
                        echo 'class="active"';
                    }
                    ?>><a href="./Index.php">Home <span
                    class="sr-only">(current)</span></a></li>
                    <li <?php
                    if ($page == "map.php") {
                        echo 'class="active"';
                    }
                    ?>><a href="./map.php">Map</a></li>


                    <?php
                    if (isset($_SESSION["id"])) {
                        ?>

                        <li <?php
                        if ($page == "itineraire.php") {
                            echo 'class="active"';
                        }
                        ?>><a href="./itineraire.php">My routes</a></li>
```

```
                            <?php if ($_SESSION["role"] == 1) { ?>
                            <li <?php
                            if ($page == "manageUsers.php") {
                                echo 'class="active"';
                            }
                            ?>><a href="./manageUsers.php">Manage users</a></li>

                            <li <?php
                            if ($page == "manageMoto.php") {
                                echo 'class="active"';
                            }
                            ?>><a href="./manageMoto.php">Manage motorcycles</a></li>
                            <?php } ?>
                    </ul>


                    <ul class="nav navbar-nav navbar-right">
                        <li><a href="./logout.php">Log Out</a></li>
                    </ul>
                    <ul class="nav navbar-nav navbar-right">
                        <li><a>Logged as <?php echo $_SESSION["username"]; ?></a></li>
                    </ul>
                    <?php
                } else {
                    ?>
                    </ul>
                    <ul class="nav navbar-nav navbar-right">
                        <li <?php
                        if ($page == "Inscription.php") {
                            echo 'class="active"';
                        }
                        ?>><a href="./Inscription.php">Sign in</a></li>

                    </ul>
                    <ul class="nav navbar-nav navbar-right">
                        <li <?php
                        if ($page == "connexion.php") {
                            echo 'class="active"';
                        }
                        ?>  ><a href="#" data-toggle="modal" data-target="#modalConnexion" onclick='$("#msgCont").attr("Hidden", "true")'>Connection</a></li>

                    </ul>
                <?php } ?>
            </div>
        </div>
    </nav>

    <?php
}

```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : contains all modal
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

//Fenêtre modal de connexion
function includeConnexionModal() {
    ?>
    <div class="modal fade" id="modalConnexion" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <div class="alert alert-danger" id='msgCont' hidden="true">
                        <strong>Error !</strong> <label id='msg'></label>
                    </div>
                    <h5 class="modal-title" >Connexion</h5>
                    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
                <div class="modal-body">
                    <div class="form-group row">
                        <label  class="col-2 col-form-label">Username</label>
                        <div class="col-4">
                            <input class="form-control" type="text"  id="username" name="username" required>
                        </div>
                    </div>
                    <div class="form-group row">
                        <label  class="col-2 col-form-label">Password</label>
                        <div class="col-4">
                            <input class="form-control" type="password"  id="pwd" name="password" required>
                        </div>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                    <button type="button" class="btn btn-primary" onclick='connexion([$("#username").val(), $("#pwd").val()])'>Connexion</button>
                </div>
            </div>
        </div>
    </div>
    <?php
}

//fenêtre modale d'ajout de moto
function AddMotorcycleModal() {
        ?>
        <div class="modal fade" id="modalMotorcycle" tabindex="-1" role="dialog" aria-labelledby="Modal motorcycle" aria-hidden="true">

            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <div class="alert alert-danger" id='msgCont' hidden="true">
                            <strong>Erreur !</strong> <label id='msg'></label>
                        </div>
                        <h5 class="modal-title" >Add a Motorcycle</h5>
                        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                            <span aria-hidden="true">&times;</span>
                        </button>
                    </div>
                    <div class="modal-body">
                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Brand</label>
                            <div class="col-4">
                                <input class="form-control" type="text"  id="AddBrand" name="Brand" required>
                            </div>
                        </div>
                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Model</label>
                            <div class="col-4">
                                <input class="form-control" type="text"  id="AddModel" name="Model" required>
                            </div>
                        </div>
                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Year</label>
                            <div class="col-4">
                                <select  id="Addyear" name="year" required>
                                    <?php
                                    for ($i = 1900; $i <= date("Y"); $i++) {
                                        echo "<option value=\"$i\">$i</option>";
                                    }
                                    ?>
                                </select>
                            </div>
                        </div>
                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Consumption</label>
                            <div class="col-4">
                                <input class="form-control" type="number" id="AddConsumption" name="Consumption" required>
                            </div>
                        </div>

                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Tiredness</label>
                            <div class="col-4">
                                <input class="form-control" type="number" id="AddTiredness" name="Tiredness" required>
                            </div>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                    <button type="button" class="btn btn-primary" onclick='AddMotorcycle($("#AddBrand").val(), $("#AddModel").val(), $("#Addyear").val(), $("#AddConsumption").val(), $("#AddTiredness").val())'>Add</button>
                </div>
            </div>
        </div>
    </div>
    <?php
}

//fenêtre modal d'import de trajet
function importerTrajetModal() {
    ?>
    <div class="modal fade" id="modalimport" tabindex="-1" role="dialog" aria-labelledby="modalimport" aria-hidden="true">
        <form method="post" action="./Includes/ImportGPX.php" enctype="multipart/form-data">
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <div class="alert alert-danger" id='msgCont' hidden="true">
                            <strong>Erreur !</strong> <label id='msg'></label>
                        </div>
                        <h5 class="modal-title" >Importer fichier .GPX</h5>
                        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                            <span aria-hidden="true">&times;</span>
                        </button>
                    </div>
                    <div class="modal-body">
                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Fichier</label>
                            <div class="col-4">
                                <input class="form-control" type="file" id="GpxFile" name="GpxFile" required>
                            </div>
                            <label class="col-2 col-form-label" >This route contains highways </label> <input class="col-lg-offset-1" type="checkbox" name="highway">
                        </div>

                        <div class="modal-footer">
                            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                            <input type="submit" class="btn btn-primary" value="Importer" name="import">
                        </div>
                    </div>
                </div>
            </div>
        </form>
    </div>
    <?php
}

//fenetre modale de confirmation de suppression d'une moto
function confirmationDeleteMotorcycleModal($id, $brand, $model, $year) {
    ?>
    <div class="modal fade" id="modaldelete<?php echo $id ?>" tabindex="-1" role="dialog" aria-labelledby="modaldelete<?php echo $id ?>" aria-hidden="true">
        <form method="post" action="./Includes/deleteMotorcycle.php" >
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">

                        <h5 class="modal-title" >Delete a motorcycle</h5>
                        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                            <span aria-hidden="true">&times;</span>
                        </button>
                    </div>
                    <div class="modal-body">
                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Are you sure to delete :</label>
                            <label  class="col-2 col-form-label"><?php echo "$brand, $model, $year" ?></label>
                            <input hidden value="<?php echo $id ?>" name="toDelete"/>
                        </div>

                        <div class="modal-footer">
                            <button type="button" class="btn btn-primary" data-dismiss="modal">Close</button>
                            <input type="submit" class="btn btn-danger" value="Yes" name="delete">
                        </div>
                    </div>
                </div>
            </div>
        </form>
    </div>
    <?php
}

//fenêtre modal de création d'une nouvelle route
function newRouteModal() {
    ?>
    <div class="modal fade" id="newRoutemodal" tabindex="-1" role="dialog" aria-labelledby="newRoutemodal" aria-hidden="true">
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <div class="alert alert-danger" id='msgCont' hidden="true">
                            <strong>Erreur !</strong> <label id='msg'></label>
                        </div>
                        <h5 class="modal-title" > New route</h5>
                        <button type="button" class="close" data-dismiss="modal"
                        aria-label="Close">
                            <span aria-hidden="true">&times;</span>
                        </button>
                    </div>
                    <div class="modal-body">
                        <div class="form-group row">
                            <label  class="col-2 col-form-label">Route name : </label>
                            <div class="col-4">
                                <input class="form-control" type="text" id="newRouteName" name="newRouteName" required>
                            </div>
                            <label class="col-2 col-form-label" >This route contains highways </label> <input class="col-lg-offset-1" type="checkbox" name="highway">
                        </div>

                        <div class="modal-footer">
                            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                            <input type="submit" class="btn btn-primary" value="Importer" name="import" onclick="CreateRoute($('#newRouteName').val(),$('#chkHighway').is(':checked'));">
                        </div>
                    </div>
                </div>
            </div>
    </div>
    <?php
}

```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : Connexion file
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

session_start();
include_once './functions.php';

if (!empty($_POST['username'])) {
    if (!empty($_POST['password'])) {
        $connexion = ConnexionUser($_POST['username'], sha1($_POST['password']));
        if (!$connexion) {
            $reponse_array['status'] = 'error';
            $reponse_array['message'] = 'wrong username or password';
        } else {
            if ($_SESSION["role"] == 3) {
                session_destroy();
                $reponse_array['status'] = 'error';
                $reponse_array['message'] = 'You are banned';
            } else {
                $reponse_array['status'] = 'success';
            }
        }
    } else {
        $reponse_array['status'] = 'error';
        $reponse_array['message'] = 'password is required';
    }
} else {
    $reponse_array['status'] = 'error';
    $reponse_array['message'] = 'username is required';
}

header('Content-type: application/json');
echo json_encode($reponse_array);

```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description :sing in fonction
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

require_once './functions.php';

//check all input values
if (isset($_POST["submit"])) {
    if (isset($_POST["username"])&&!empty($_POST["username"])) {
        $values["username"] = $_POST["username"];
        if (isset($_POST["email"])&&!empty($_POST["email"])) {
            $values["email"] = $_POST["email"];
            if (isset($_POST["password"])&&!empty($_POST["password"])) {
                $values["password"] = sha1($_POST["password"]);
                if (isset($_POST["password2"])&&!empty($_POST["password2"])) {
                    $values["password2"] = sha1($_POST["password2"]);
                    if (isset($_POST["brand"])&&!empty($_POST["brand"])) {
                        $values["brand"] = $_POST["brand"];
                        if (isset($_POST["model"])&&!empty($_POST["model"])) {
                            $values["model"] = $_POST["model"];
                            if (isset($_POST["year"])&&!empty($_POST["year"])) {
                                $values["year"] = $_POST["year"];

                                if ($values["password"] == $values["password2"]) {
                                    if(strlen($values["username"])>=5){
                                        signin(json_encode($values));
                                          header("location: ../Index.php?validation=Inscription%20réussie");
                                    }else {
                                    redirect("Username length minimum 5 characters", $values);
                                }
                                } else {
                                    redirect("Password aren't the same", $values);
                                }
                            } else {
                                redirect("Year is required", $values);
                            }
                        } else {
                            redirect("model is required", $values);
                        }
                    } else {
                        redirect("brand is required", $values);
                    }
                } else {
                    redirect("Password confirmation is required", $values);
                }
            } else {
                redirect("Password is required", $values);
            }
        } else {
            redirect("Email is required", $values);
        }
    } else {
        redirect("Username is required");
    }
}else {
        redirect();
    }

exit();

function redirect($error = NULL, $values = NULL) {
    $values = json_encode($values);
    header("location: ../Inscription.php?error=$error&values=$values");
}
?>

```

```php
<?php

/*
 * Author : Lucien Camuglia
 * Description : Import a GPX file
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

/**
 * Connexion à la base de données
 * @staticvar PDO $pdo  objet de connexion
 * @return PDO  connexion à la base
 */
function connexionDb() {

    try {
        //database login infos
        $serveur = '127.0.0.1';
        $pseudo = 'fyr';
        $pwd = 'NekoFyr';
        $db = 'findyourride';

        static $pdo = null;

        if ($pdo === NULL) {
            // Conection to database.
            $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
            $pdo = new PDO("mysql:host=$serveur;dbname=$db", $pseudo, $pwd, $pdo_options);
            $pdo->exec("Set Character set UTF8");
        }
        return $pdo;
    } catch (Exception $exc) {
        echo $exc->getTraceAsString();
    }
}

$pdo = connexionDb();

/**
 * Prépare et éxécute une requête
 * @param string $query requête a ééxécuter
 * @param tableau $params  éventuel paramètre pour la requête
 * @return PDOStatement
 */
Function PrepareExecute($query, $params = NULL) {
    global $pdo;
// Préparation de la requête SQL.
    $st = $pdo->prepare($query);
// Execution de la requête SQL.
    $st->execute($params);
    return $st;
}

/**
 * Connecte un utilisateur
 * @param String $User Nom d'utilisateur
 * @param Sha1_String $Password Mot de passe cripté
 * @Return Boolean Vrai si la connexion a marché
 */
function ConnexionUser($User, $Password) {
    global $pdo;

    $query = 'SELECT idUser, Username, Password, role FROM users ';

    $st = PrepareExecute($query);
    $connecter = false;

    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        if ($data['Username'] == $User && $data['Password'] == $Password) {
            $id = $data['idUser'];
            $username = $data["Username"];
            $role = $data["role"];

            $connecter = true;
            break;
        }
    }
    if ($connecter) {
        $_SESSION["role"] = $role;
        $_SESSION["id"] = $id;
        $_SESSION["username"] = $username;
        return true;
    } else {
        return false;
    }
}

/**
 * Return all brand
 * @return array all motorcycle brand
 */
function GetMotorcycleBrand() {
    $query = "select distinct Brand from moto";
    $st = PrepareExecute($query);
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = $data["Brand"];
    }
    return $array_response;
}

/**
 * Return all motorcycle model
 * @return array with all motorcycle model
 */
function GetAllMotorcycleModel() {
    $query = "select distinct model from moto order by model";
    $st = PrepareExecute($query);
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = $data["model"];
    }
    return $array_response;
}
```

```php

/**
 * return all years of motorcycle
 * @return array with all years
 */
function GetAllMotorcycleYear() {
    $query = "select distinct year from moto order by year";
    $st = PrepareExecute($query);
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = substr($data["year"], 0, 4);
    }
    return $array_response;
}

/**
 * return all consumptions of motorcycle
 * @return array with all consumptions
 */
function GetAllMotorcycleConsumption() {
    $query = "select distinct consumption from moto order by consumption";
    $st = PrepareExecute($query);
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = $data["consumption"];
    }
    return $array_response;
}

/**
 * return all Tiredness of motorcycle
 * @return array with all Tiredness
 */
function GetAllMotorcycleTiredness() {
    $query = "select distinct Tiredness from moto order by Tiredness";
    $st = PrepareExecute($query);
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = $data["Tiredness"];
    }
    return $array_response;
}

/**
 * Return all motorcycle
 * @return array with all motorcycle ordered by brand
 */
function GetAllMotorcycles() {
    $query = "select * from moto order by Brand";
    $st = PrepareExecute($query);
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = $data;
    }
    return $array_response;
}

/**
 * Sign in a user
 * @param array $values
 * @return int as boolean  false if the user exists and the id if not
 */
function signin($values) {
    $values = json_decode($values);
    //get the user in the bd who have the same username
    $query = "Select Username FROM users where Username=:username;";
    $params = array('username' => $values->username);
    $st = PrepareExecute($query, $params);
    //if something is returned, the user already exists
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        return false;
    }
    //get the id moto of the selected motorcycle
    $query = "SELECT idMoto FROM moto WHERE Brand=:brand AND model=:model AND year=:year";
    $params = array(
        'brand' => $values->brand,
        'model' => $values->model,
        'year' => $values->year . "-01-01"
    );
    $st = PrepareExecute($query, $params);
    $idMoto = $st->fetch(PDO::FETCH_ASSOC)["idMoto"];
    //insert in the db the new user
    $query = "INSERT INTO users(Username, Password, email, idMoto, role) VALUES (:username,:password,:email,:idmoto,2)";
    $params = array(
        'username' => $values->username,
        'password' => $values->password,
        'email' => $values->email,
        'idmoto' => $idMoto
    );
    $st = PrepareExecute($query, $params);

    global $pdo;
    //return the last inserted id
    return $pdo->lastInsertId();
}

/**
 * Get all routes
 * @param $idUser facultative user id
 * @return JSON array
 */
function GetRoutes($idUser = NULL) {
    if ($idUser == NULL) {
        $query = "select * from route natural join users  order by RouteName ";
        $st = PrepareExecute($query);
    } else {
        $query = "select * from route where idUser = :iduser order by RouteName";
        $params = array(':iduser' => $idUser);
        $st = PrepareExecute($query, $params);
    }
    $array_response = [];
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = $data;
    }
    return $array_response;
}
```

```php

/**
 * Create a new route
 * @param string $name name of the trip
 * @param int $iduser
 * @param boolean $containsHighway
 * @return int Route Id
 */
function CreateRoute($name, $iduser, $containsHighway) {
    $query = "INSERT INTO route(RouteName, idUser, highway) VALUES (:name,:user,
    :highway)";
    $params = array(
        'name' => $name,
        'user' => $iduser,
        'highway' => $containsHighway
    );
    $st = PrepareExecute($query, $params);
    global $pdo;
    return $pdo->lastInsertId();
}

/**
 * Deleate all points of a road
 * @param int $idRoute
 */
function deleatePlaces($idRoute) {
    $query = "DELETE FROM `place` WHERE idRoute = :idRoute";
    $params = array(
        'idRoute' => $idRoute
    );
    $st = PrepareExecute($query, $params);
}

/**
 * Add a new point of a road
 * @param double $lat latitude
 * @param double $lon longitude
 * @param int $position position of the point in the route
 * @param int $idRoute
 */
function addPlaceToRoute($lat, $lon, $position, $idRoute) {
    $query = "INSERT INTO place(latitude, longitude, position, idRoute) VALUES
    (:lat,:lon,:pos,:route)";
    $params = array(
        'lat' => $lat,
        'lon' => $lon,
        'pos' => $position,
        'route' => $idRoute
    );
    $st = PrepareExecute($query, $params);
}

/**
 * Save a GPX file in the database
 * @param string $file path and filname
 * @param int $iduser
 * @param boolean $containsHighway
 * @return int routeID
 */
function Gpx2Sql($file, $iduser, $containsHighway) {
    //create a new dom Document
    $dom = new DOMDocument();
    //open the file
    $dom->load($file);
    //get the route name in the file
    $names = $dom->getElementsByTagName("name");
    $name = $names->item(0)->nodeValue;
    //get all trakpoint
    $trackpoint = $dom->getElementsByTagName("trkpt");
    if ($trackpoint == NULL) {
        //get all route point
        $trackpoint = $dom->getElementsByTagName("rtept");
    }
    //create the route and get the id
    $idRoute = CreateRoute($name, $iduser, $containsHighway);
    $position = 0;
    //add all points in the database
    foreach ($trackpoint as $point) {
        $lat = $point->getAttribute("lat");
        $lon = $point->getAttribute("lon");
        addPlaceToRoute($lat, $lon, $position++, $idRoute);
    }
    return $idRoute;
}

/**
 * Convert a path into a GPX file
 * @param string $name road name
 * @param JSON array $path
 * @return string filename
 */
function Path2Gpx($name, $path) {
    $path = json_decode($path);
    $file = $name . ".gpx";
    if (file_exists($file)) {
        unlink($file);
    }
    $handle = fopen($file, "w");
    $string = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>";
    fwrite($handle, $string);
    $string = "<gpx version=\"1.0\" creator=\"FindYourRide.org\"
    xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
    xmlns=\"http://www.topografix.com/GPX/1/0\"
    xsi:schemaLocation=\"http://www.topografix.com/GPX/1/0
    http://www.topografix.com/GPX/1/0/gpx.xsd\">";
    fwrite($handle, $string);
    $string = "<trk> <name>$name</name> <trkseg>";
    fwrite($handle, $string);
    foreach ($path as $point) {
        $string = "<trkpt lat=\"" . $point->lat . "\" lon=\"" . $point->lng . "\">
        </trkpt>";
        fwrite($handle, $string);
    }

    $string = "</trkseg> </trk> </gpx>";
    fwrite($handle, $string);
    fclose($handle);
    return $file;
}

/**
 * Delete a motorcycle
 * @param int $id motorcycle id
 */
function deleteMotorcycle($id) {
    $query = "Delete from moto where idMoto=:id";
    $params = array(
        'id' => $id
    );
    $st = PrepareExecute($query, $params);
}

/**
 * Get a user and his motorcycle
 * @return JSON array values
 */
function getUsersNMotorcycle() {
    $array_response = [];
    $query = "Select * from users natural join moto;";
    $st = PrepareExecute($query);
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array_response[] = $data;
    }
    return $array_response;
}

/**
 * Get the role of an user
 * @param int $id user id
 * @return int role of the user
 */
function getUserRoleById($id) {
    $query = "Select role from users where idUser=:id;";
    $params = array(
        'id' => $id
    );
    $st = PrepareExecute($query, $params);
    $data = $st->fetch(PDO::FETCH_ASSOC);
    return $data["role"];
}

/**
 * Add sinuosity of a road
 * @param int $idRoute
 * @param int $sinueusite road sinuosity
 */
function AddSinuosity($idRoute, $sinueusite) {
    $query = "Update route SET Sinuosity=:sinuosity where idRoute=:idRoute;";
    $params = array(
        'sinuosity' => $sinueusite,
        'idRoute' => $idRoute
    );
    PrepareExecute($query, $params);
}

/**
 * Add elevation of a road
 * @param int $idRoute
 * @param int $elevation road elevation
 */
function AddElevation($idRoute,$elevation) {
    $query = "Update route SET Slope=:elevation where idRoute=:idRoute;";
    $params = array(
        'elevation' => $elevation,
        'idRoute' => $idRoute
    );
    PrepareExecute($query, $params);
}

/**
 * Add length of a road
 * @param int $idRoute
 * @param int length road length
 */
function AddLength($idRoute,$length) {
    $query = "Update route SET Length=:length where idRoute=:idRoute;";
    $params = array(
        'length' => $length,
        'idRoute' => $idRoute
    );
    PrepareExecute($query, $params);
}

/**
 * Get the most sinuous road
 * @return int sinuosity value
 */
function GetMostSinuousRoad() {
    $query = "SELECT max(Sinuosity)*10000 AS Road FROM route;";
    $st = PrepareExecute($query);
    $val = $st->fetchAll(PDO::FETCH_ASSOC);
    return (int) $val[0]["Road"];
}

/**
 * Get the less sinuous road
 * @return int sinuosity value
 */
function GetLessSinuousRoad() {
    $query = "SELECT min(Sinuosity)*10000 AS Road FROM route;";
    $st = PrepareExecute($query);
    $val = $st->fetchAll(PDO::FETCH_ASSOC);
    return (int) $val[0]["Road"];
}

/**
 * Get the most steepest road
 * @return int steepest value
```

```php
 */
function GetMostSteepestRoad() {
    $query = "SELECT max(Slope) AS Road FROM route;";
    $st = PrepareExecute($query);
    $val = $st->fetchAll(PDO::FETCH_ASSOC);
    return (int) $val[0]["Road"];
}
/**
 * Get the less steepest road
 * @return int steepest value
 */
function GetLessSteepestRoad() {
    $query = "SELECT min(Slope) AS Road FROM route;";
    $st = PrepareExecute($query);
    $val = $st->fetchAll(PDO::FETCH_ASSOC);
    return (int) $val[0]["Road"];
}

?>
```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : contains all function used in ajax
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

session_start();
include './functions.php';

if (isset($_REQUEST["fonction"])) {
    $function = $_REQUEST["fonction"];

    switch ($function) {
        case "GetModel" : GetModel($_GET["Brand"]);
            break;
        case "GetYear" : GetYear($_GET["Brand"], $_GET["Model"]);
            break;
        case "UserExists" : UserExists($_GET["Username"]);
            break;
        case "GetMotorcycles" : GetMotorcycles($_GET["Brand"], $_GET["Model"], $_GET["Year"], $_GET["Consumption"], $_GET["Tiredness"]);
            break;
        case"GetRoutePoints":GetRoutePoints($_GET["idRoute"]);
            break;
        case "SaveNewRoute" : SaveNewRoute($_REQUEST["idRoute"], $_REQUEST["route"], $_REQUEST["sinuosite"], $_REQUEST["elevation"], $_REQUEST["length"]);
            break;
        case "AddMotorcycle" : AddMotorcycle($_GET["Brand"], $_GET["Model"], $_GET["Year"], $_GET["Consumption"], $_GET["Tiredness"]);
            break;

        case "UpdateUserRole" : UpdateUserRole($_GET["idUser"], $_GET["Role"]);
            break;
        case "GetUserRoleById" : GetUserRole($_GET["idUser"]);
            break;
        case "Download" : downloadRoute($_REQUEST["name"], $_REQUEST["path"]);
            break;
        case "FilterRoad" : FilterRoad($_GET["sinuosity"], $_GET["slope"], $_GET["highway"], $_GET["time"]);
            break;
        case "GetRoutes" : GetRoutesJSON();
            break;
        case "GetRoadsInfos" : GetRoadsInfos($_GET["idRoute"]);
            break;
        case "CreateRoute" : CreateNewRoute($_GET["name"], $_GET["containsHighway"]);
            break;
        case "DeleteRoute" : DeleteRoute($_GET["idRoute"]);
            break;
        default : exit();
            break;
    }
}else{
    header('location: ../index.php');
}
exit();
```

```php

    //Retrun motorcycle models by brand
    function GetModel($brand) {
        $query = "select distinct model from moto where Brand = :brand ";
        $params = array('brand' => $brand);
        $st = PrepareExecute($query, $params);
        while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
            $array_response[] = $data["model"];
        }
        echo json_encode($array_response);
    }

    //Retrun motorcycle's year by brand and model
    function GetYear($brand, $model) {
        $query = "select distinct year from moto where Brand = :brand  and model = :model order by year";
        $params = array('brand' => $brand, 'model' => $model);
        $st = PrepareExecute($query, $params);
        while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
            $array_response[] = substr($data["year"], 0, 4);
        }
        echo json_encode($array_response);
    }

    //check if an user exists
    function UserExists($username) {
        $query = "select idUser from users where Username = :username";
        $params = array('username' => $username);
        $st = PrepareExecute($query, $params);
        $array_response["exist"] = false;
        while ($data = $st->fetch(PDO::FETCH_ASSOC))
            $array_response["exist"] = true;

        echo json_encode($array_response);
    }

    //Return all motorcycles according to input criteria
    function GetMotorcycles($brand, $model, $year, $consumption, $tiredness) {
        $query = "Select * from moto where Brand like :brand and model like :model and year like :year and consumption like :consumption and Tiredness like :tiredness order by Brand";
        $params = array('brand' => $brand, 'model' => $model, 'year' => $year, 'consumption' => $consumption, 'tiredness' => $tiredness);
        $st = PrepareExecute($query, $params);
        $array_response = "";
        while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
            $array_response[] = $data;
        }
        echo json_encode($array_response);
    }

    //Get all points of a road
    function GetRoutePoints($idRoute) {
        $query = "Select * from place where idRoute=:idRoute order by position;";
        $params = array('idRoute' => $idRoute);
        $st = PrepareExecute($query, $params);
        $array_response = "";
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array["Latitude"] = $data["latitude"];
        $array["Longitude"] = $data["longitude"];
        $array_response[] = $array;
    }

    echo json_encode($array_response);
}

//save a new route
function SaveNewRoute($idroute, $route, $sinueusite, $elevation, $length) {
    //delete all places from this road
    deleatePlaces($idroute);

    AddSinuosity($idroute, $sinueusite);
    AddElevation($idroute, $elevation);
    AddLength($idroute, $length);
    $position = 0;
    $fromweb = false;
    if (is_string($route)) {
        $route = json_decode($route);
        $fromweb = true;
    }
    foreach ($route as $point) {
        if ($fromweb) {
            $lat = $point->lat;
            $lon = $point->lng;
        } else {
            $lat = $point["latitude"];
            $lon = $point["longitude"];
        }
        addPlaceToRoute($lat, $lon, $position++, $idroute);
    }
}

//add a new motorcycle
function AddMotorcycle($brand, $model, $year, $consumption, $tiredness) {
    $array_response = [];
    $array_response["error"]["status"] = false;
    $array_response["error"]["message"] = "";

    if (!empty($brand) && !empty($model) && !empty($year) && !empty($consumption) && !empty($tiredness)) {
        $query = "Insert Into moto (Brand, model, year, consumption, Tiredness) values (:brand, :model, :year, :consumption, :tiredness);";
        $params = array('brand' => $brand, 'model' => $model, 'year' => $year . '-01-01', 'consumption' => $consumption, 'tiredness' => $tiredness);
        PrepareExecute($query, $params);
    } else {
        $array_response["error"]["status"] = true;
        $array_response["error"]["message"] = "Please fill all field";
    }

    echo json_encode($array_response);
}

//update a user role
```

```php
function UpdateUserRole($idUser, $idRole) {
    $array_response = [];
    $array_response["error"]["status"] = false;
    $array_response["error"]["message"] = "";

    //check if the user is admin
    if (isset($_SESSION["role"]) && $_SESSION["role"] == 1) {

        $query = "UPDATE users SET role = :role WHERE idUser=:idUser;";
        $params = array('role' => $idRole, 'idUser' => $idUser);
        $st = PrepareExecute($query, $params);
        $Newrole;
        switch (getUserRoleById($idUser)) {
            case 1: $Newrole = "Administrator";
                break;
            case 2: $Newrole = "User";
                break;
            case 3: $Newrole = "Ban";
                break;
            default : $Newrole = "error";
                break;
        }
        $array_response["datas"]["NewRole"] = $Newrole;
    } else {
        $array_response["error"]["status"] = true;
        $array_response["error"]["message"] = "Rights needed";
    }

    echo json_encode($array_response);
}

//get user role by id
function GetUserRole($idUser) {
    $result;
    switch (getUserRoleById($idUser)) {
        case 1: $result = "Administrator";
            break;
        case 2: $result = "User";
            break;
        case 3: $result = "Ban";
            break;
    }
    echo json_encode($result);
}

//download prepare to download and return filename
function downloadRoute($name, $path) {
    $file = Path2Gpx($name, $path);
    echo $file;
}

//return a filtered road
function FilterRoad($sinuosity, $slope, $highway, $time) {
    $query = "Select * from route natural join users where Sinuosity <= :sinuosity AND Slope <= :slope AND Time <= :time AND highway = :highway ORDER BY RouteName;";
    // $query  = "Select * from route where Sinuosity <= 135 AND Slope <= 0 AND Time <= 1 AND highway = 0;";
    $params = array(
        "sinuosity" => $sinuosity / 10000,
        "slope" => $slope,
        "time" => $time,
        "highway" => ($highway == "true") ? '1' : '0');
    $st = PrepareExecute($query, $params);
    $array_response = [];
    while ($data = $st->fetch(PDO::FETCH_ASSOC)) {
        $array["RouteName"] = $data["RouteName"];
        $array["idRoute"] = $data["idRoute"];
        $array["Username"] = $data["Username"];
        $array_response[] = $array;
    }

    echo json_encode($array_response);
}

//return all routes in JSON
function GetRoutesJSON() {
    $routes = GetRoutes();
    echo json_encode($routes);
}

//get roads info
function GetRoadsInfos($idRoute) {
    $query = "Select * from route where idRoute = :idRoute";
    $params = Array("idRoute" => $idRoute);
    $st = PrepareExecute($query, $params);
    $datas = $st->fetch(PDO::FETCH_ASSOC);
    $array_response["Length"] = $datas["Length"] / 1000;
    $array_response["Highway"] = $datas["highway"];
    $array_response["Time"] = $datas["Time"];
    $array_response["Sinuosity"] = (int) (($datas["Sinuosity"] * 10000 / GetMostSinuousRoad()) * 10);
    $array_response["Slope"] = round($datas["Slope"] / GetMostSteepestRoad() * 10);

    //if th euser is connected, return motorcycle consumption if not, return 0
    if (isset($_SESSION["id"])) {
        $query = "select * from moto natural join users where idUser = :idUser ";
        $params = Array("idUser" => $_SESSION["id"]);
        $st = PrepareExecute($query, $params);
        $datas = $st->fetch(PDO::FETCH_ASSOC);
        $motorcycleConsumption = $datas["consumption"];
    } else {
        $motorcycleConsumption = 0;
    }

    $array_response["MotorcycleConsumption"] = (int) $motorcycleConsumption;

    echo json_encode($array_response);
}

//add a route in database and return id
function CreateNewRoute($name, $containsHighway) {
    $id = CreateRoute($name, $_SESSION["id"], $containsHighway);
    echo json_encode($id);
}
```

```php

//delete a route
function DeleteRoute($idRoute) {
    $query = "Delete from place where idRoute=:idRoute;";
    $params = array("idRoute" => $idRoute);
    PrepareExecute($query, $params);

    $query = "Delete from route where idRoute=:idRoute;";
    PrepareExecute($query, $params);

    echo json_encode("Success");
}

```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : Delete a motorcycle
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

session_start();
include_once './functions.php';

if(isset($_SESSION["role"]) && $_SESSION["role"]==1){
    if(isset($_POST["delete"])){
        if(isset($_POST["toDelete"])){
            deleteMotorcycle($_POST["toDelete"]);
            header('location: ../manageMoto.php');
            exit();
        }
    }
}

 header('location: ../index.php');

exit();
```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : Download a file
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */

if (isset($_GET['file'])) {
    $file = $_GET['file'];
    //verify if file exists
    if (file_exists($file) && is_readable($file)) {
        //prepare to download
        header('Content-Type: application/octet-stream');
        header("Content-Disposition: attachment; filename=\"$file\"");
        //download the file
        readfile($file);
        //delete it
        unlink($file);
    } else {
        //display an 404 error
        header("HTTP/1.0 404 Not Found");
        echo "<h1>Error 404: File Not Found: <br /><em>$file</em></h1>";
    }
}


```

```
 */
include_once './functions.php';
```

```php
<?php
/*
 * Author : Lucien Camuglia
 * Description : Import a GPX file
 * Date : April-june 2017
 * Version : 1.0 LC BaseVersion
 */
include_once './functions.php';
session_start();
//verify if import has been pressed
if (isset($_POST["import"])) {
    //veriy if a file is selected
    if (isset($_FILES["GpxFile"])) {
        //verify if the file format is gpx
        if (pathinfo($_FILES["GpxFile"]["name"])["extension"] == "gpx") {
            //set if the road contains highway
            $containsHighway = isset($_POST["highway"]);
            // save the file on the server
            $idRoute = Gpx2Sql($_FILES["GpxFile"]["tmp_name"], $_SESSION["id"], $containsHighway);
            //redirect the user
             header("location: ../itineraire.php?NewIti&id=$idRoute");
        } else {
            header("location: ../itineraire.php?modal=modalimport&error=extension%20GPX%20nécéssaire");
        }
    }
}
```