ÉCOLE CENTRALE LYON

Projet d'Étude n° 82

Application Web dediée à la recherche des places de stationnement

Auteurs :
M. Antoine EDY
M. Nathan BAUERLÉ
M. Gabin CALMET
M. Thibaut DEJEAN
M. Raffi SARKISSIAN
M. Arthur BRUNO

Encadrants :
M. Phillipe MICHEL
M. Abdelmalek ZINE
M. Alexandre SAIDI
Mme Veronique BILLAT
M. Fabio ANTONIALLI
M. Alexis GIAUQUE
(président de jury)

Remis le 3 juin 2022

Table des matières

Table des figures

Remerciements

Nous voudrions d'abord remercier les tuteurs techniques de ce projet M. Phillipe Michel, M. Abdelmalek Zine et M. Alexandre Saidi, ainsi que les tuteurs de gestion de projet et de communication M^{me} Veronique Billat et M. Fabio Antonialli respectivement, pour nous avoir conseillé et guidé tout au long du projet. Antoine EDY aimerait également remercier les fondateurs du site OpenClassrooms (openclassrooms.com), Mathieu Nebra et Pierre Dubuc, qui mettent en ligne des ressources gratuites pour apprendre différents langages de programmation. C'est grâce à ces outils accessibles que de tels projets peuvent voir le jour.

Resumé

Quel conducteur n'a jamais souhaité savoir en temps réel si une place de stationnement à l'autre bout de la ville est libre ou non ? Le problème de stationnement est un casse-tête pour les urbanistes depuis des décennies. La solution : un outil informatique permettant à chacun de savoir où se garer à un moment donné. L'une des clefs de la réalisation de ce projet est la recherche de données en temps réel sur le remplissage des parkings de Lyon. Ce Projet d'Études a pour but de créer un site internet complet et intuitif, qui répond à plusieurs besoins. Ainsi, ce site doit permettre aux lyonnais de trouver rapidement le parking le plus proche de leur destination. Il fournira également des informations en direct sur les places qui y sont disponibles, en tenant compte des préférences personnelles du client (places destinées aux personnes à mobilité réduite, places moto ou vélo...). Enfin, ce site internet gratuit oriente l'utilisateur vers un site GPS afin de le guider vers le parking sélectionné.

Abstract

Every driver has always wanted to know if a parking lot on the other side of town is available or not. The "parking issue" has been a headache for city planners for decades. The solution : a simple website allowing everyone to know where to park at a given time. One of the keys to the realization of this project is the research of real time data on the filling of the parking lots in Lyon. The aim of this project is to create a complete and intuitively usable website that meets several needs. This site will allow Lyon residents to quickly find the parking lot

closest to their destination. It will also provide live information on the available spaces, taking into account the customer's personal preference (parking space reserved for handicapped drivers, motorcycle park place...). Finally, this free website directs the user to a Global Positioning System website in order to guide the user to the selected parking lot.

1 Introduction

Le présent Projet d'Études répond à la problématique imposée par les enseignants - tuteurs, de concevoir un outil informatique pour aider à trouver des places de stationnement.

La création d'un site internet qui permet à l'utilisateur, en suivant le cahier des charges établi, de faciliter la recherche de places dans Lyon fut la solution adoptée. Notre étude se limitera géographiquement à Lyon et son agglomération.

2 Description du projet

2.1 Contexte du projet

La recherche de stationnement est aujourd'hui l'une des principales causes des émissions de CO_2 et des nuisances sonores dans les grandes villes. Limiter le temps engagé par la population dans la recherche d'un lieu pour garer leur voiture permettra également de réduire la production de stress des conducteurs. À Lyon, plus de 434 heures sont perdues chaque jour à cause du temps de recherche des parkings et 67% des conducteurs ont déjà abandonné un voyage dû à l'impossibilité de trouver une place disponible. Les exigences sur la recherche de stationnement sont également plus pointues. En effet, la variété des moyens de transport demande une grande diversité de places. Les motos, les vélos, mais aussi les voitures électriques et les camionnettes peuvent être cités.

2.2 Le projet

Ce Projet d'études consiste en la conception d'un site internet qui vise à faciliter le stationnement dans la ville de Lyon, ainsi que faciliter la recherche de places spécifiques (handicap, points de charge des véhicules électrique, etc. ...). Ce site doit être rapide d'utilisation et intuitif pour l'utilisateur, il faudra donc minimiser le temps de passage sur le site pour chaque client et lui donner les informations utiles à sa recherche.

2.3 Services concurrents

Ce type de service est aujourd'hui assez courant et le principe de ce projet n'est pas fondamentalement novateur. Il existe cependant certains points encore jamais explorés. Cette partie s'intéresse aux sites et applications qui existent

déjà et qui proposent aussi le service de recherche de stationnement. Le but est d'avoir un aperçu global de la concurrence pour se démarquer en essayant de proposer quelque chose de nouveau.

Quatre applications et/ou sites internet ont été sélectionnés. Leurs réponses à différents critères sont répertoriées dans la figure 1. Cette comparaison permet d'identifier les directions à prendre pour se démarquer de la concurrence.

Nom	Site	App	Gratuit	GPS inclut	Type de stationnement	Temps réel	Fonctionnalités intéressantes	Remarques
Yespark	Oui	Oui	Non	Via Google	Parking privés	Oui	- Borne de recharge - Places doubles - Beaucoup de choix de taille du véhicule - Informations précises sur les parkings	- L'abonnement est cher
Parclick	Oui	Oui	Oui	X	Parking privés et payants	Non	- Avis et notes sur les parkings	- Peu de choix de parking
Polly	Non	Oui	Oui	Oui	Parkings payants	Non	- Rappel de la localisation de la voiture	- Ne marche bien qu'à Paris
Parkopedia	Oui	Oui	Oui	Via Google	Tous	Non	- Parkings en voirie (parcmètres) - Parkings réservés aux clients d'un établissement - Affichage du prix sur la carte interactive - Signalement d'erreurs	- Présence discrète de publicité sur le site - Marche en Europe et en Amérique du Nord

FIGURE 1 – Caractéristiques de quelques services concurrents

Parkopedia est le site qui parait le plus complet car il répertorie les places gratuites et payantes, les parkings publics et privés, et même les places en voirie équipées d'un parcmètre. Cette dernière fonctionnalité est assez rare et c'est une réelle valeur ajoutée de Parkopedia par rapport à ses concurrents. Cependant, sur ce site, l'utilisateur ne dispose d'aucun moyen pour savoir si un parking contient une place libre ou non.

Au contraire le service **Yespark** répertorie certes un nombre très limité de parkings privés mais a un accès direct au nombre de places disponibles dans chaque parking. L'avantage est qu'il est possible de louer une place de parking pour une longue durée (1 mois) mais le prix de location dépasse régulièrement les 100€.

Polly est du même style que Yespark sauf que cette application contient un guidage par GPS directement dans l'application sans passer par Google Maps. Polly guide l'utilisateur vers des parkings payants et lui permet de réserver sa place à l'avance. Cependant ce service est indisponible via un site internet et ne marche que par application mobile.

Enfin, **Parkclick** est aussi du même style que Yespark et Polly sauf qu'en plus

de ne pas proposer de parkings gratuits, de ne pas avoir accès à la disponibilité des parkings, il n'y a aucun moyen pour l'utilisateur d'accéder au guidage par GPS en un clic. Pour être guidé il doit copier l'adresse du parking et la coller sur Google Maps.

Actuellement, il n'existe pas sur le marché de service proposant à la fois la disponibilité des parkings en temps réel et un accès aux parkings publics. Cette direction sera donc privilégiée afin de se démarquer des autres services.

Pour conclure cette partie, la figure 2 récapitule les positions et les choix pris vis à vis des principaux concurrents.

Nom	Site	App	Gratuit	GPS inclut	Type de stationnement	Temps réel	Fonctionnalités intéressantes	Remarques
Yespark	Oui	Oui	Non	Via Google	Parking privés	Oui	- Borne de recharge - Places doubles - Beaucoup de choix de taille du véhicule - Informations précises sur les parkings	- L'abonnement est cher
Parclick	Oui	Oui	Oui	X	Parking privés et payants	Non	- Avis et notes sur les parkings	- Peu de choix de parking
Polly	Non	Oui	Oui	Oui	Parkings payants	Non	- Rappel de la localisation de la voiture	- Ne marche bien qu'à Paris
Parkopedia	Oui	Oui	Oui	Via Google	Tous	Non	- Parkings en voirie (parcmètres) - Parkings réservés aux clients d'un établissement - Affichage du prix sur la carte interactive - Signalement d'erreurs	- Présence discrète de publicité sur le site - Marche en Europe et en Amérique du Nord
Trouve Ta Place	Oui	Non	Oui	Via Google	Parking publics ou privés	Oui	- Borne de recharge - Place handicapées	- Ne fonctionne qu'en local

FIGURE 2 – Comparaison de notre projet avec les services concurrents

2.4 Objectifs principaux

Les objectifs principaux de ce projet sont :

- Établir le design du site et l'implémenter dans sa structure afin d'obtenir un site agréable et intuitif.
- Élaborer un algorithme permettant de trouver le parking le plus approprié à la destination de l'utilisateur ainsi qu'à ses besoins (places vélos, motos...) en temps réel.
- Implémenter une carte interactive permettant à l'utilisateur de visualiser les parkings les plus proches de lui.
- Re-diriger l'utilisateur vers un site GPS afin de le guider jusqu'au parking sélectionné.

La fonctionnalité de disponibilité en temps réel restreint notre étude aux parkings principalement souterrains ou au moins possédant un contrôle des flux de voitures. En effet, il est indispensable de connaître à un instant donné le nombre précis de places disponibles dans le parking.
Nous avons privilégié un site à une application ; le site est accessible depuis son smartphone ainsi que son ordinateur, ce qui n'est pas le cas de l'application.

2.5 Objectifs secondaires

Si les objectifs principaux sont réalisés et fonctionnels, le projet peut être amélioré à l'aide de fonctionnalités supplémentaires qui sont décrites sous forme d'objectifs secondaires dans cette liste :

- Intégrer les places en voiries à la recherche de places disponibles. Cette fonctionnalité nécessite une étude probabiliste de la disponibilité des places dans les rues ainsi qu'une cartographie très précise de Lyon et de son agglomération. Cela n'a pas été fait cette année, mais est un point intéressant sur lequel travailler dans la suite du projet.
- Ajouter de nouveaux critères d'option pour la recherche de parkings (bornes de recharge pour véhicules électriques, places handicapés...). Nous avons partiellement mis en place cette fonctionnalité. La prochaine étape serait de trouver une base de données qui contient le prix des parkings, donnée intéressante pour l'utilisateur.
- Ajouter un système de connexion afin d'accélérer la recherche des utilisateurs réguliers. Cela n'a pas été mis en place par manque de temps.

2.6 Budget

Comme le budget est exclusivement dû au temps que nous avons passé à travailler sur ce projet, voici une liste de l'estimation de la durée réelle des différentes tâches.

- Apprentissage des langages de programmation : 20 h
- Premier rendez-vous de suivi de projet : 15 h
- Structure du site, conception et programmation : 25 h
- Extraction des données : 35 h
- Programmation de l'algorithme de score : 50 h
- Deuxième rendez-vous de suivi de projet : 15 h

- Finalisation (correction des bogues, ajout de fonctionnalités secondaires) : 40 h
- Rédaction du rapport final : 20 h

Au total, 220 heures de travail ont été comptabilisées, avec un salaire de 20€/h le budget de masse salariale s'élèverait à 4400€.

Ce projet n'a pas eu de dépenses externes, donc le budget total de ce projet est aussi de 4400€.

3 Parcours utilisateur

Il est commun lors de la présentation d'une application web de commencer par présenter le parcours utilisateur, c'est à dire le parcours typique d'un utilisateur quelconque qui souhaite utiliser le service proposé par le site internet.

Pour ce genre d'application purement utilitaire, le temps de connexion d'un utilisateur moyen sur le site doit être faible : une des qualités du site internet doit être sa facilité d'utilisation. Dès lors, la page d'accueil doit être celle qui permet à l'utilisateur d'effectuer sa recherche. C'est pourquoi la barre de recherche se situe directement sur la page d'accueil comme l'illustre la figure 3. Le bouton cliquable "Options" affiche trois options de recherche supplémentaires pour les places de personnes à mobilité réduite, les places motos et les places vélos. En cliquant sur le bouton "Valider" et en patientant le temps que les calculs s'effectuent, on accède à la page du résultat de la recherche.

On arrive ensuite sur cette page de résultat (figure 4) : à gauche les dix parkings les plus adaptés à la recherche, à droite la localisation de ces parkings par rapport à la destination. L'étape suivante est éventuellement de cliquer sur la voiture situer à la droite de chaque parking pour accéder à l'itinéraire de sa position jusqu'au parking en question. On arrive alors sur le site `https://www.google.com/maps/`, puis on suit l'itinéraire affiché.

Ainsi s'achève l'itinéraire d'un utilisateur lambda.

4 Récupération des données et mise en place de la base de données

La récupération des données liées aux parkings est une étape très importante dans le projet puisque le site est entièrement basé sur celle-ci. Comme les données

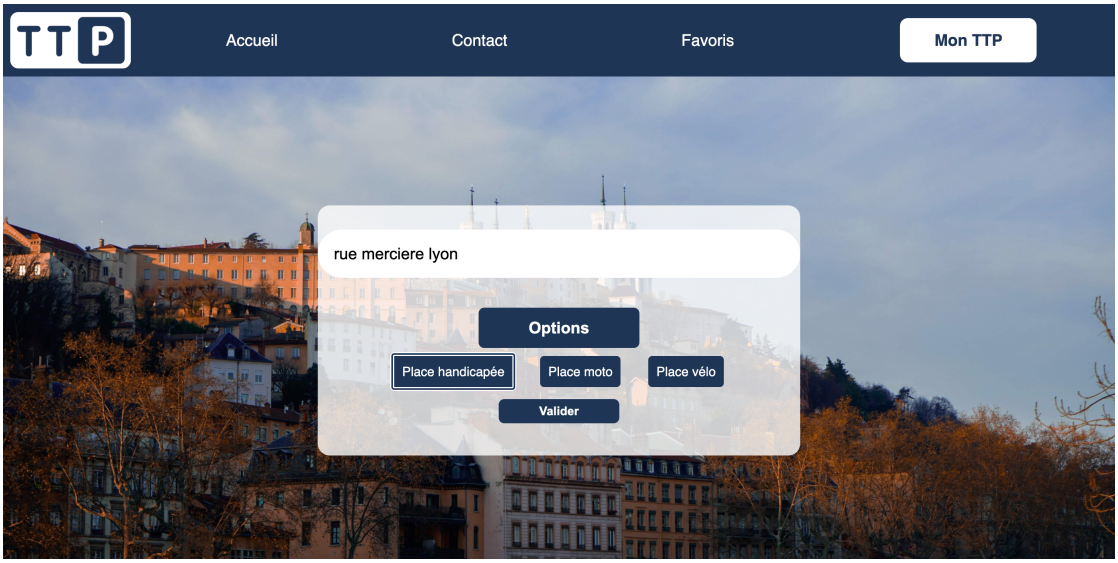

FIGURE 3 – Page d'accueil du site

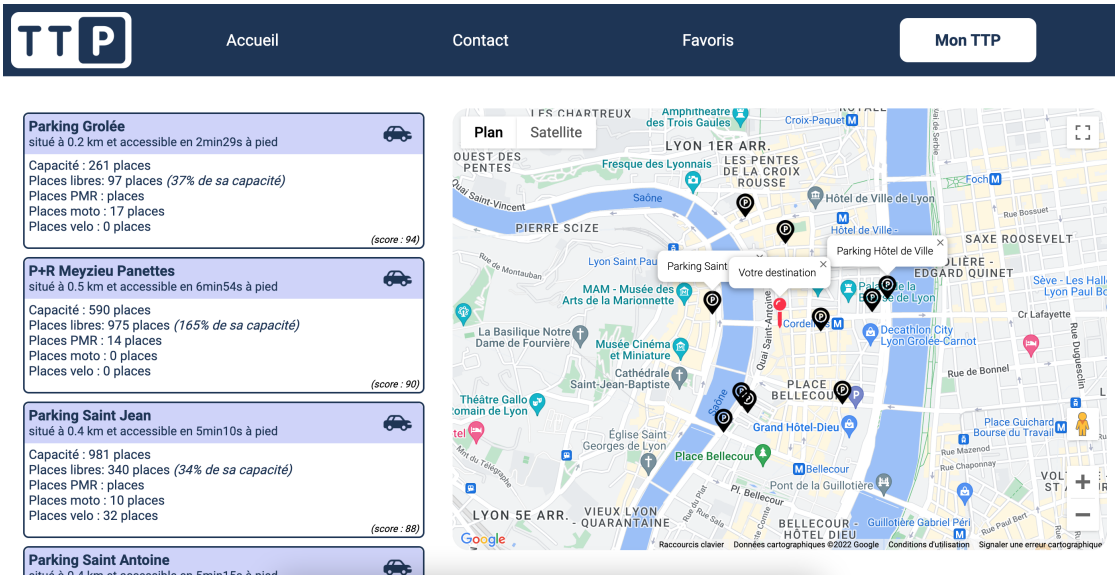

FIGURE 4 – Page des résultats du site

sont récupérées en temps réel, la qualité des calculs dépend immédiatement de la qualité de la base de données. C'est le site Data Grand Lyon qui a été choisi pour un accès gratuit à des centaines de bases de données sur la métropole de Lyon.

4.1 Données des parkings en temps réel

La première étape dans la construction de la base de données est de trouver des statistiques en temps réel des parkings de Lyon. Pour cela, la table intitulée "Parkings de la Métropole de Lyon - disponibilités temps réel" a été sélectionnée depuis le site Data Grand Lyon. Cette base de données donne accès en temps réel aux places libres dans 101 parkings publics de Lyon. Ces places libres sont réparties en quatre classes : la capacité voiture, la capacité moto, la capacité vélo et la capacité PMR (places pour les personnes à mobilité réduite). Ces quatre classes permettent à l'utilisateur une personnalisation de la recherche de stationnement et elles nous permettent de lui proposer des résultats plus pertinents. Cette table de données donne également accès au nom du parking ainsi qu'à son identifiant unique dans la base, au gestionnaire du parking et à l'identifiant du gestionnaire.

L'utilisation de cette base de données très complète fournie par le site Data Grand Lyon nécessite la signature préalable d'une licence. L'acquisition de cette licence nécessite une description du projet afin que Data Grand Lyon prenne connaissance de nos besoins et décide d'accorder ou non l'accès aux données. Finalement l'accès à l'entièreté des données présentes a été accordé pour le projet, sous certaines conditions précisées dans la licence. Les données peuvent être utilisées pendant un an et cela est reconductible 2 fois soit une durée maximale de 3 années. La licence se trouve en annexe pour avoir plus de détails sur ces conditions.

Plusieurs problématiques apparaissent dès lors :

- La date de la dernière actualisation étant précisée avec les données, un premier problème est apparu ici. Certaines données de parkings sont actualisées assez peu régulièrement, à plus d'une heure d'intervalle. Ce problème inhérent à la base de données choisie ne va pas pouvoir être résolu dans ce projet.
- Un autre point problématique est que les coordonnées géographiques des parkings ne sont pas indiquées dans les données. Or l'algorithme du site internet compare et calcule des distances. Il faut donc trouver les coordonnées de ces parkings.

4.2 Données des coordonnées des parkings

Une première solution afin d'obtenir les coordonnées des parkings est de trouver une autre base de données en temps réel plus complète possédant également les positions géographiques. Cependant, il n'existe aucune base de données de la sorte fonctionnant en temps réel.

En revanche, la position des parkings n'évoluant pas avec le temps, il n'est pas nécessaire d'utiliser une base de données évoluant en temps réel pour obtenir les données géographiques des parkings. Une base de donnée créée par les mêmes créateurs que celle sélectionnée précédemment et qui possède les coordonnées géographiques des parkings a été choisie. C'est donc cette deuxième base de données qui a été utilisée en parallèle de la première. Cette base est entièrement libre et a pu être utilisée sans passer par une licence.

4.3 Mise en place de la base de données

Afin de faciliter l'utilisation de ces données, une base de donnée sqlite est créée. Les données récupérées précédemment sont donc placées dans cette base. Cependant, par souci d'utilisation dans le code, un fichier Json est créé à partir de cette base de données. C'est donc ce fichier Json qui va être lu par le code en pHp.

5 Structure du site

Le but de ce site est qu'il soit simple d'utilisation et intuitif car ce site se rend utile si la recherche de place est rapide et sans difficulté. Ainsi la structure du site est importante pour que l'utilisateur ne se perde pas entre les pages.

5.1 Header et footer

Premièrement on retrouve le header et le footer qui seront présents sur toutes les pages.

Le header se situe en haut de la page et contient le logo du site, ainsi que 4 boutons : le premier est un bouton accueil qui permet de revenir à la page d'accueil depuis n'importe quelle page du site. on retrouve deux autres boutons "Contact" et "Favoris" qui permettent d'accéder à des pages que nous détaillerons

plus tard. Le dernier bouton "Mon TTP" permet de se connecter ou de créer un compte si c'est notre première fois sur le site. La connexion sur ce site permettra à l'utilisateur d'aller encore plus vite dans ses recherches en enregistrant ses destinations et ses parkings préférés.

FIGURE 5 – Header du site

On retrouve ensuite le footer en bas de page. Il contient les éléments essentiels d'un site qui sont les mentions légales ainsi qu'un bouton permettant d'accéder à la description du site.

FIGURE 6 – Footer du site

5.2 Page de garde

Comme expliqué précédemment, afin de rendre le site simple d'utilisation, la recherche de parkings est accessible directement sur la page de garde. La barre de recherche est même la première chose visible du site. Cette barre de recherche est accompagnée d'un menu d'options permettant à l'utilisateur d'affiner sa recherche. Le menu d'option est pour l'instant composé de filtres pour les vélos, les motos et les places PMR.

La page de garde se poursuit avec 3 boutons. Le premier est un bouton "Favoris" permettant d'accéder aux destinations et parkings enregistrés si l'utilisateur a créé un compte. Le deuxième est un bouton permettant l'accès à la carte des parkings de Lyon. Le dernier est un bouton "Trafic" qui mène sur une page résumant les différents problèmes rencontrés dans les parkings dans Lyon (travaux, fermeture...).

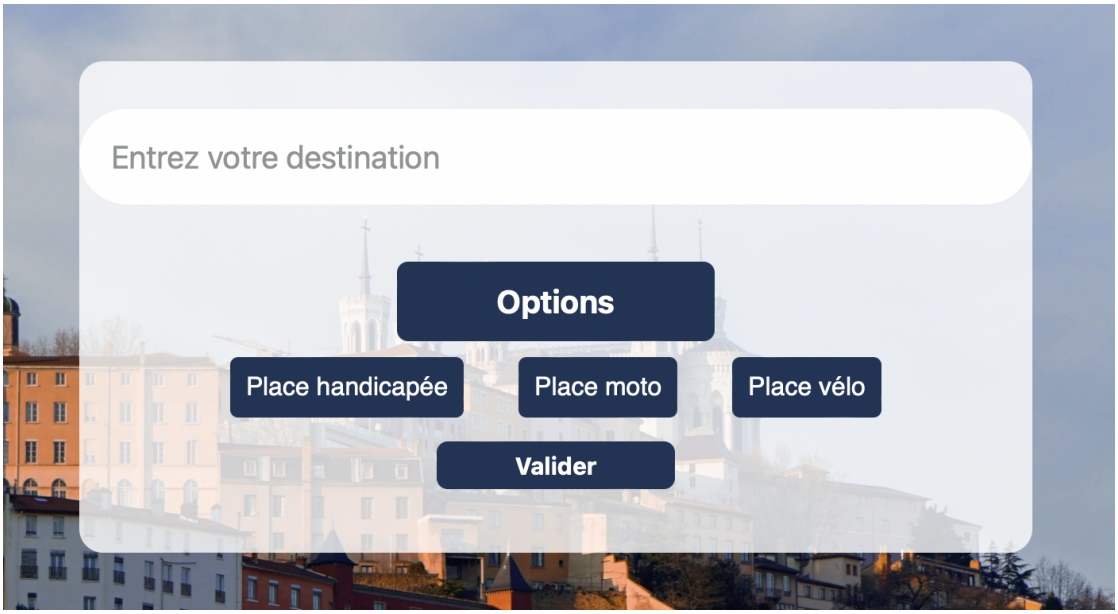

FIGURE 7 – Barre de recherche et options

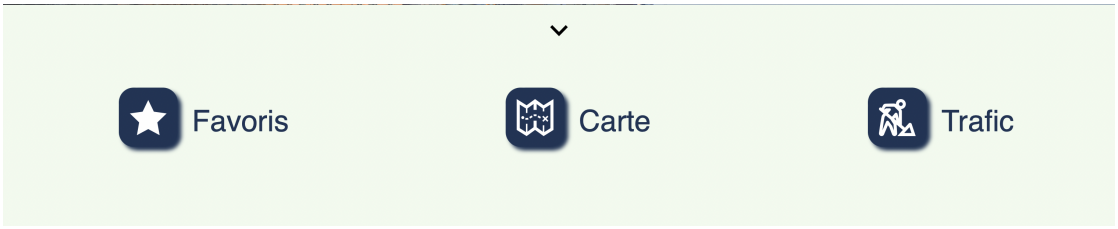

FIGURE 8 – Suite de la page de garde du site

5.3 Page de résultat de la recherche

Cette page permet à l'utilisateur de visualiser les résultats de la recherche. Elle contient les informations nécessaires pour que l'utilisateur fasse un choix de parking.

La structure de cette page ainsi que le code pour y parvenir sont présentés dans la partie algorithmique du rapport.

5.4 Pages supplémentaires

Sur ce site sont présentes différentes page permettant d'améliorer l'expérience utilisateur.

Dans un premier temps, la page "Contacts" permet à l'utilisateur de contacter

les propriétaires du site afin de signaler une erreur ou de faire une remarque quelconque.

FIGURE 9 – Cadre permettant d'envoyer un message au créateur du site

Dans un second temps, la page "Mon TTP" permet à l'utilisateur de se connecter afin d'avoir accès à des favoris qui seront présents sur la page "Favoris".

FIGURE 10 – Cadre permettant de se connecter

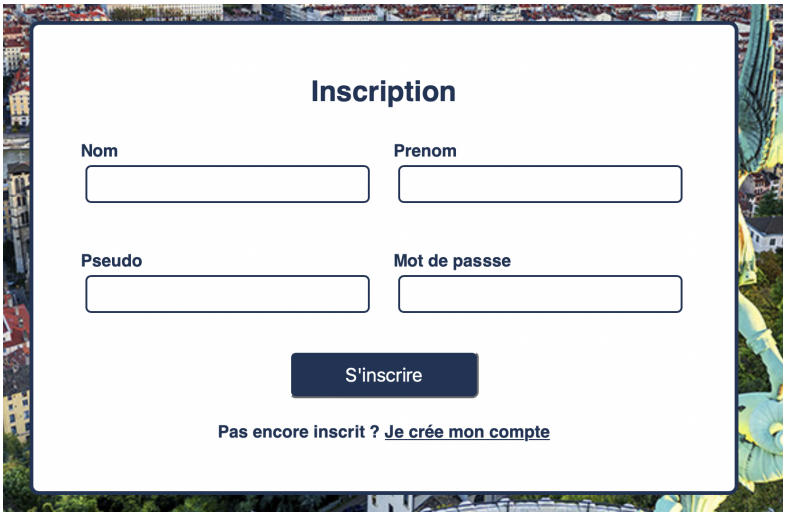

FIGURE 11 – Cadre permettant de s'inscrire sur le site

6 Utilisation des interfaces de programmation d'application (API)

L'ambition de ce projet de donner les disponibilités d'un parking en temps réel nécessite d'avoir accès à des données extérieures. Le moyen le plus simple d'accéder à ces données est par l'intermédiaire des API, ou interfaces de programmation d'application. Dans notre cas, l'API est une interface logicielle qui permet de transmettre des données d'un logiciel tiers à notre site internet.

Les API utilisées dans le projet fonctionnent comme suit : une requête est envoyée au service, qui prend comme paramètres la clef API ainsi que les éléments propres à la requête, qui sont détaillés ici. Quatre APIs sont nécessaires afin de mener à bien notre projet :

Data Grand Lyon : disponibilités en temps réel des parkings dans Lyon et son agglomération. La licence est à retrouver en annexe.

$\longrightarrow$ Ici seule la clef API est nécessaire, aucun autre paramètre n'est renseigné : nous cherchons à récupérer l'ensemble des données disponibles sur les parkings de Lyon et de son agglomération.

Google Map - Distance Matrix : calcul de distance et de temps de parcours à pieds entre deux points. Le service est payant mais nous utilisons les 400€ d'utilisation offerts pendant 60 jours à la création d'un compte Google Cloud.

$\longrightarrow$ En plus de la clef API, nous renseignons le point de départ (l'adresse d'un parking), le point d'arrivée (la destination finale de l'utilisateur) et le mode de trajet (à pieds en l'occurrence).

Google Map - Maps JavaScript : permet d'intégrer une carte personnalisée et interactive au site. Le service est payant mais nous utilisons les 400€ d'utilisation offerts pendant 60 jours à la création d'un compte Google Cloud.

$\longrightarrow$ En plus de la clef API, nous renseignons le centrage de la cart,e le niveau de zoom, la position des différents repères présents ainsi que leur nom et leur icon.

Google Map - Geocode : permet de trouver les coordonnées d'un lieu en fonction de son nom. Le service est payant mais nous utilisons les 400€

d'utilisation offerts pendant 60 jours à la création d'un compte Google Cloud.

⟶ En plus de la clef API, nous renseignons le nom du lieu.

7 Partie algorithmique

Cette partie est consacrée au "back-end" du projet, c'est à dire la partie algorithmique avec laquelle l'utilisateur ne peut pas intéragir. Par souci de lisibilité, l'ensemble du code n'est pas retranscrit ici mais est disponible dans son intégralité en annexe.

Lorsque l'utilisateur valide sa recherche (adresse de la destination et options pour les places), les données sont envoyées à une page `algorithme.php` dédiée à l'algorithme. Le mode d'envoi choisi est le `GET` : les paramètres choisis par l'utilisateur se trouvent dans l'url. Elle est donc de la forme suivante :

```
http://localhost:8080/algorithme.php?lieu=rue+merciere+lyon&pmr=0&moto
=0&velo=0
```

Ces paramètres sont ainsi récupérés à l'aide de variables en utilisant la commande `$_GET`, donc par exemple `$_GET['pmr']=1` signifie que l'utilisateur a coché la case PMR, et l'algorithme va privilégier les parkings possédant de nombreuses places pour les personnes à mobilité réduite.
L'essence de l'algorithme et les éléments permettant sa compréhension sont présentés dans cette partie.

7.1 La récupération des données fournies par les API

Les lignes de codes permettant de récupérer les données fournies par les APIs sont très différentes en fonction de celles considérées. Elles sont détaillées dans cette partie.

7.1.1 L'API Data Grand Lyon

Les premières données a récupérer sont celles des parkings. Il est nécessaire en effet d'avoir les informations suivantes :

— Leur nom,

— Le nombre de places disponibles,

— Leur capacité,
— Le nombre de places PMR, vélos et motos.

Pour vérifier que la licence a bien été validée et que les données sont accessibles, il faut préciser un identifiant ainsi qu'un mot de passe pour faire notre requête. Ce système n'utilise donc pas ces fameuses "clefs API" classiques. Les données sont récupérées comme suit :

```
$username='gabin.calmet@ecl21.ec-lyon.fr';
// Ici notre mot de passe est caché...
$password='********';
$URL='https://download.data.grandlyon.com/ws/rdata/
    pvo_patrimoine_voirie.pvoparkingtr/all.json?maxfeatures=-1&start=1
    ';

// Un nouvel objet cUrl est initialisé
$ch = curl_init();

// Tout est une option avec PHPCurl !
curl_setopt($ch, CURLOPT_URL,$URL);

// Permet de pouvoir récupérer le résultat
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);

// On autorise les requêtes
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);

curl_setopt($ch, CURLOPT_USERPWD, "$username:$password");

$result=curl_exec ($ch);
$parsed_json = json_decode($result);
```

`$parsed_json` contient donc les données de l'ensemble des parkings en format Json. Son exploitation sera étudiée dans la prochaine partie.

7.1.2 L'API Google Map - Distance Matrix

L'objectif est ici de récupérer la distance et la durée à pieds entre chaque parking et la destination de l'utilisateur. Ces données sont conservées respectivement en mètre et en seconde dans les variables `$distance` et `$duration`.

```
\\ On cache notre clef !
$key ='****************';
$lien ='https://maps.googleapis.com/maps/api/distancematrix/json?
    departure_time=' . 'now' . '&destinations=' . $lieu . '&origins='
```

```
    . $latitude . '%2C' . $longitude . '&mode=walking' . '&key=' .
    $key;
    $json1 = file_get_contents($lien);
    $parsed_json1 = json_decode($json1);
    foreach($parsed_json1->{'rows'} as $p1) :
        $duration = $p1->{'elements'}[0]->{'duration'}->{'value'};
        $distance = $p1->{'elements'}[0]->{'distance'}->{'text'};
        $_SESSION['duration']=$duration;
        $_SESSION['distance']=$distance;
    endforeach;
```

7.1.3 L'API Google Map - Geocode

Ici la destination est transformée (la chaine de caractères stockée dans `$lieu`) en coordonnées GPS, qui sont stockées dans des variables `$_SESSION`, choix qui est explicité dans la partie suivante.

```
$lien2 = 'https://maps.googleapis.com/maps/api/geocode/json?address='
    . $lieu . '&key=***********';

$json3 = file_get_contents($lien2);
$parsed_json3 = json_decode($json3);
foreach($parsed_json3->{'results'} as $p3) :
    $lat_d = $p3->{'geometry'}->{'location'}->{'lat'};
    $long_d = $p3->{'geometry'}->{'location'}->{'lng'};
    $_SESSION['lat_d']=$lat_d;
    $_SESSION['long_d']=$long_d;
endforeach;
```

Ces valeurs vont permettre de placer un repère "Votre destination" sur la carte de la page des résultats.

7.1.4 L'API Google Map - JavaScript embed

Les lignes de codes correspondantes seront détaillées dans la partie Affichage des résultats - la carte interactive.

7.2 L'attribution d'un score aux parkings

L'algorithme va boucler sur l'ensemble des parkings disponibles dans la base de données pour leur attribuer un score, détaillé ci-dessous. Le score est divisé en deux parties : la première concerne les places disponibles ainsi que les options choisies par l'utilisateur (PMR, vélo, moto). Le score est calculé par la formule suivante, qui donnera un score entre 0 et 1.

```
$score_places = (1-((10000-$nb)/10000) + (1-((100-$capa_pmr)/100))*
    $_GET['pmr'] + (1-((100-$capa_moto)/100))*$_GET['moto'] +
    (1-((100-$capa_velo)/100))*$_GET['velo'])/(1+$_GET['pmr']+$_GET['
    moto']+$_GET['velo']);
```

La variable $nb est définie comme suit :

```
foreach($parsed_json->{'values'} as $p) :
    $etat = $p->{'etat'};
endforeach;
$nb = explode(' ', $etat, 2);
$nb=(int) $nb[0];
```

Elle contient donc le nombre de places disponibles en temps réel dans le parking.

La seconde partie concerne la distance, ou plus précisément la durée du trajet à pieds entre la destination et les parkings voisins. La durée du trajet est récupérée comme suit :

```
foreach($parsed_json1->{'rows'} as $p1) :
    $duration = $p1->{'elements'}[0]->{'duration'}->{'value'};
endforeach;
```

Le score de la durée est ensuite calculé :

```
$score_duration = (15000-$duration)/15000;
```

Le score final est la somme des deux scores précédents à un coefficient près devant le score de la durée. Ce coefficient permet de pondérer les deux scores. En l'occurrence dans cet algorithme, la durée de déplacement mise entre le parking et la destination est 7 fois plus importante que le score de place. ce score est

multiplié par 100 puis on soustrait 600 pour avoir un score compris entre 0 et 100. Ces constantes ont été choisies empiriquement.

```
$score = ($score_places + 7*$score_duration)*100-600;
```

Le score final étant calculé, on peut classer les parkings et proposer à l'utilisateur une liste des parkings les plus adaptés à sa demande.

Cette méthode `_GET` permet de passer les variables de la page d'accueil à celle de l'algorithme. Afin de conserver des données plus longtemps et entre les différentes pages du site, il est intéressant d'ouvrir une session php. Les sessions sont un moyen simple de stocker des données individuelles pour chaque utilisateur. Les variables de la forme `$_SESSION` se conservent tant que l'utilisateur ne s'est pas déconnecté du site. Ainsi, toutes les informations sur les parkings sont stockées dans une liste qui sera associée plus tard à une variable de ce type :

```
$one_p = [$nom,$gestionnaire,$nb, $capa_voiture, $capa_pmr,
    $capa_moto, $capa_velo, $score, $distance, $duration, $longitude,
    $latitude];
$L[] = $one_p;
```

`$L` contient alors la liste des informations de tous les parkings ainsi que les variables `$distance` et `$duration` associées au parking correspondant.

Cette liste est ainsi triée en fonction du score obtenu par chaque parking :

```
$taille = count($L);
for($i = 0; $i < $taille; $i++)
{
    for($j = $taille-1; $j >= $i; $j--)
    {
      if($L[$j+1][7]> $L[$j][7])
      {
          $temp = $L[$j+1];
          $L[$j+1] = $L[$j];
          $L[$j] = $temp;
      }
  }
}
```

La liste `$L` est donc triée par ordre décroissant des scores. Enfin, comme indiqué précédemment :

```
$_SESSION['L']=$L;
```

Il suffira alors, dans n'importe quelle page mais surtout dans celle qui affiche les résultats, de retrouver les données pour les afficher avec `<?php echo($L[$i][$j])?>` en faisant varier `$j`.

8 Affichage des résultats

La page est divisée en deux parties :
— À gauche (1) la liste des dix parkings les plus adaptés
— À droite (2) leur emplacement sur une carte sur laquelle est également affichée le repère de destination

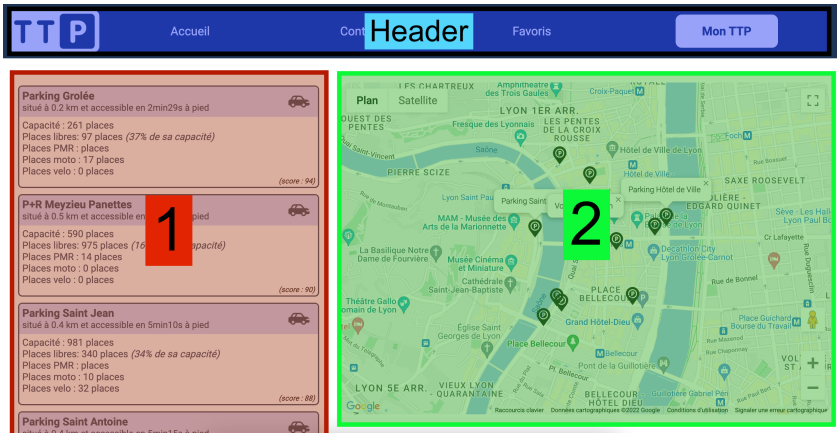

FIGURE 12 – Division de la page des résultats

8.1 La liste des parkings

Les dix premiers parkings de la liste précédemment triée sont affichés. Un bouton cliquable (la voiture bleue foncée) mène l'utilisateur vers une page Google Maps qui a comme destination le parking choisi. La structure HTML de ces blocs est la suivante :

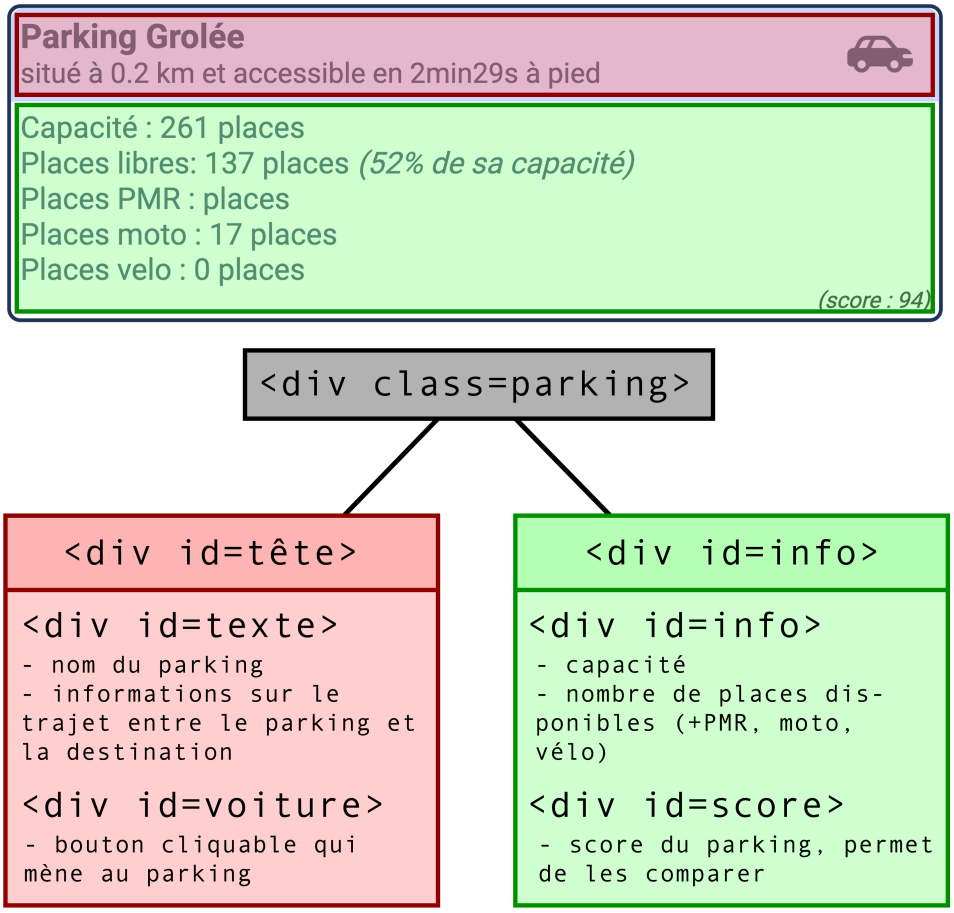

FIGURE 13 – Division du bloc parking

```
<div class = "parking">
  <div class="tete">
    <div id="texte">
      <p id="name"><?php echo($L[$i][0])?></p>
      <p id="name2"><?php echo ('situé à '); echo($L[$i][8]); echo('
  et accessible en '); echo($minute); echo('min'); echo($seconde);
  echo('s à pied');?></p>
    </div>
    <div id=voiture>
      <a href=<?php echo("'https://www.google.fr/maps/dir//" . $L[$i
  ][11] . ',' . $L[$i][10] . "'" ) ?> target="_blank">
        <img id="destination" src="images/auto.png"></a>
    </div>
  </div>
```

```
    <div  id="info">
    </br>
  </br>
  <p>Capacité : <?php echo($L[$i][3])?> places</br>
    Places libres: <?php echo($L[$i][2])?> places <i>(<?php echo(
   intval(($L[$i][2]/$L[$i][3])*100)) ?>% de sa capacité)</i></br>
    Places PMR : <?php echo($L[$i][4])?> places</br>
    Places moto : <?php echo($L[$i][5])?> places</br>
    Places velo : <?php echo($L[$i][6])?> places
    <div id='score'> <i> (score : <?php echo(floor($L[$i][7]))?>)</i>
    </div></p>
  </div>
</div>
```

8.2 La carte interactive

L'API Google Maps - Maps in Java utilise comme son nom le laisse deviner du JavaScript. Ce code est a placé entre les bornes <script> et </script> dans le fichier pHp. La carte est initialisée de cette manière :

```
let map;

function initMap() {
map = new google.maps.Map(document.getElementById("map"), {
  center: new google.maps.LatLng('<?php echo $_SESSION['lat_d'];?>',
    '<?php echo $_SESSION['long_d'];?>'),
  zoom: 15,
});
```

L'ensemble des repères sur le plan vont ensuite être placés. La constante `features` est défini.Elle contient la latitude et la longitude des repères, que l'on retrouve avec la liste `$L` définie précédemment.

```
 const features = [
  {
    position: new google.maps.LatLng('<?php echo $L[0][11];?>', '<?
   php echo $L[0][10];?>'),
    type: "parking",
  },
```

```
  {
    position: new google.maps.LatLng('<?php echo $L[1][11];?>', '<?
   php echo $L[1][10];?>'),
    type: "parking",
  },
  .
  .
  .
  // Ceci pour les dix parkings
```

Après avoir défini une liste `data` qui contient les noms des parkings, une nouvelle variable est définie :

```
var infowindow = [
{
vaar: new google.maps.InfoWindow({content: data[0]})
},
{
vaar: new google.maps.InfoWindow({content: data[1]})
},
.
.
.
\\ Ceci pour les dix parkings
```

Le marqueur correspondant à la destination finale de l'utilisateur est ensuite placé. Une icône particulière lui est attribuée, `icon2`, qui est rouge, pour le différencier des parkings.

```
const marker = new google.maps.Marker({
    position: features[10].position,
    icon: icon2,
    map: map,
  });
  google.maps.event.addListener(marker, 'click', function() {
    infowindow[10].vaar.open(map,marker);
  });
```

Les dix marqueurs des parkings sont placés avec une icône noire :

```
for (let i = 0; i < features.length-1; i++) {
    const marker = new google.maps.Marker({
      position: features[i].position,
      icon: icon,
      map: map,
    });
    google.maps.event.addListener(marker, 'click', function() {
      infowindow[i].vaar.open(map,marker);
    });
  }
}

window.initMap = initMap;
```

9 Conclusion

Pour rappel, les objectifs principaux du présent Projet d'Études étaient la création d'un site internet permettant aux automobilistes de la région lyonnaise de connaître l'état de remplissage des parkings de l'agglomération pour se garer plus simplement. A la fin d'un travail d'un an, le site internet permet effectivement aux lyonnais de savoir où se garer afin d'avoir une place disponible et comment se rendre dans le parking concerné. De plus, de nombreuses informations permettent de choisir entre les différents parkings proposés, dont la distance à la destination, le remplissage ou encore le score. Enfin, les parkings recensés par le site internet sont visibles grâce à une carte interactive.

Cependant, l'actualisation de la base de données du Grand Lyon ne permet pas d'avoir des informations actualisées suffisamment régulièrement pour certains parkings. La barre de recherche pourrait également être améliorée en proposant des rues de Lyon au fur et à mesure que l'utilisateur tape sa destination. De plus, le site internet *Trouve Ta Place* ne comporte pas encore de fonction "Mon TTP" qui permet aux utilisateurs de se connecter afin d'accélérer leur recherche et la page Traffic n'est pas complète puisqu'il faudrait avoir des données sur l'état des parkings.

Ces améliorations pourraient faire l'objet d'objectifs principaux pour une poursuite du projet. L'objectif secondaire du Projet d'Études de considérer les places en voirie est également une extension très prometteuse du site. Par ailleurs, la création d'une application est une piste de travail qui semble importante afin

de faciliter l'accès au service.

Toutefois, nous sommes fiers d'avoir atteint notre objectif principal : le site est fonctionnel et l'ensemble des fonctionnalités souhaitées est disponible. De plus, le site internet permet d'obtenir des informations sur les places spécifiques (places handicapées...) disponibles dans les parkings les plus proches et permet à l'utilisateur de se diriger vers le parking.

10 Annexe

10.1 Structure complète du site internet

pe82-site

Name	Kind	Size	Date Modified
datas	Folder	78,9 MB	03/06/2022 at 10:27:46
data_test.csv	CSV Document	3 KB	16/03/2022 at 15:38:48
Parkings.csv	CSV Document	8 KB	13/05/2022 at 19:27:04
bal_200046977.csv	CSV Document	14,4 MB	16/03/2022 at 16:00:05
base.db	Document	17,0 MB	12/05/2022 at 23:35:27
pvo_patrimoine_voirie.json	Text File	968 KB	16/03/2022 at 16:00:05
Rues.json	Text File	46,4 MB	16/03/2022 at 15:38:47
Ajout_lon_lat_db.py	Text File	717 bytes	08/05/2022 at 12:00:48
base_parking.json	Text File	46 KB	16/03/2022 at 16:00:05
Parkings.json	Text File	17 KB	13/05/2022 at 19:34:16
traitement_data_rue_db.py	Text File	532 bytes	19/03/2022 at 22:28:05
client	Folder	5,8 MB	03/06/2022 at 10:29:21
submit	Folder	2 KB	16/03/2022 at 15:38:47
submit_form.php	PHP script	632 bytes	02/02/2022 at 15:21:20
submit_inscription.php	PHP script	506 bytes	16/03/2022 at 15:38:47
deconnexion.php	PHP script	90 bytes	02/02/2022 at 15:21:20
submit_connexion.php	PHP script	740 bytes	16/03/2022 at 15:38:47
dist	Folder	3 KB	26/05/2022 at 11:49:53
recherche.js	Text File	3 KB	26/05/2022 at 11:49:53
images	Folder	5,6 MB	01/06/2022 at 18:30:57
backgrounds	Folder	5,5 MB	02/02/2022 at 15:21:20
index-background.jpg	JPEG image	1,1 MB	02/02/2022 at 15:21:20
contact_background.jpg	JPEG image	3,4 MB	02/02/2022 at 15:21:20
connexion-background.jpg	JPEG image	531 KB	02/02/2022 at 15:21:20
inscription-background.jpg	JPEG image	456 KB	02/02/2022 at 15:21:20
icones	Folder	4 KB	02/02/2022 at 15:21:20
arrow-down.svg	SVG	168 bytes	02/02/2022 at 15:21:20
logo_TTP.svg	SVG	674 bytes	02/02/2022 at 15:21:20
iconmonstr-star-3.svg	SVG	217 bytes	02/02/2022 at 15:21:20
arrow-right.svg	SVG	146 bytes	02/02/2022 at 15:21:20
iconmonstr-map-10.svg	SVG	1 KB	02/02/2022 at 15:21:20
iconmonstr-construction-15.svg	SVG	2 KB	02/02/2022 at 15:21:20
voiture.gif	GIF image	31 KB	01/06/2022 at 18:23:16
auto.png	PNG image	9 KB	01/06/2022 at 18:30:57
pin.png	PNG image	2 KB	18/05/2022 at 16:59:05
parking2.png	PNG image	17 KB	18/05/2022 at 11:52:11

Name	Kind	Size	Date Modified
parking.png	PNG image	40 KB	18/05/2022 at 11:44:03
logo2.png	PNG image	7 KB	16/03/2022 at 15:05:17
w80.png	PNG image	13 KB	06/02/2022 at 12:48:48
w50.png	PNG image	13 KB	06/02/2022 at 12:46:39
includes	Folder	3 KB	16/03/2022 at 15:59:45
header.php	PHP script	2 KB	08/05/2022 at 11:32:22
footer.php	PHP script	508 bytes	19/03/2022 at 22:29:16
script	Folder	1 KB	26/05/2022 at 11:49:53
index.js	Text File	1 KB	26/05/2022 at 11:49:53
styles	Folder	12 KB	26/05/2022 at 11:49:53
map.css	Text File	179 bytes	16/03/2022 at 15:05:17
contact.css	Text File	1 KB	16/03/2022 at 15:05:17
connexion.css	Text File	2 KB	16/05/2022 at 09:36:27
index.css	Text File	4 KB	20/03/2022 at 09:34:49
main.css	Text File	282 bytes	02/02/2022 at 15:21:20
algorithme.css	Text File	55 bytes	30/05/2022 at 10:32:00
recherche.css	Text File	2 KB	26/05/2022 at 11:49:53
mentions_legales.css	Text File	549 bytes	16/03/2022 at 15:05:17
resultat_recherche.css	Text File	1 KB	01/06/2022 at 18:37:46
inscription.css	Text File	1 KB	16/03/2022 at 15:05:17
javascript	Folder	96 KB	30/05/2022 at 10:51:03
jquery.js	Text File	90 KB	30/05/2022 at 10:48:44
recherche.ts	MPEG-2 Transport Stream	3 KB	26/05/2022 at 11:49:53
connexion.php	PHP script	1 KB	16/03/2022 at 15:42:12
index.php	PHP script	3 KB	26/05/2022 at 11:49:53
contact.php	PHP script	1 KB	16/03/2022 at 15:05:17
mentions_legales.php	PHP script	3 KB	16/03/2022 at 15:05:17
test.php	PHP script	771 bytes	30/05/2022 at 11:07:29
resultat_recherche.php	PHP script	7 KB	01/06/2022 at 18:38:38
inscription.php	PHP script	2 KB	16/03/2022 at 15:05:17
algorithme.php	PHP script	4 KB	02/06/2022 at 18:31:27
recherche.php	PHP script	3 KB	26/05/2022 at 11:49:53
data_test_rue.json	Text File	7 KB	16/03/2022 at 15:38:47
index.js	Text File	925 bytes	18/05/2022 at 15:27:36
Parkings.json	Text File	17 KB	18/05/2022 at 15:04:05
favicon.ico	Windows icon image	15 KB	16/05/2022 at 06:11:56

10.2 Ensemble des codes (pHp, HTML, JavaScript) qui composent le site

On évitera de mettre les feuilles de style CSS qui ne sont pas d'une grande utilité dans la compréhension de notre projet.

10.2.1 Pages principales

École centrale de Lyon
36, Avenue Guy de Collongue
code du labo concerné
69134 Écully