

# Guide de développement



Carnel est un outil de covoiturage entre développé comme une application interne pour une petite entreprise, permettant des déplacements en commun au sein de l'IUT Réseaux et Télécommunications de Montbéliard. Ce projet a été développé en quelques jours, dans le cadre d'une Saé (Situation d'Apprentissage et d'évaluation), du lundi 9 mars au vendredi 12 mars 2022. Afin de réaliser se projet, nous travaillons essentiellement sur nos ordinateur portable personnels avec l'utilisation d'un serveur XAMPP qui contient un serveur Apache et phpmyadmin. Mais nous travaillons aussi sur les ordinateurs de l'IUT qui nous facilitait l'exportation de notre projet sur rt-serv. Nous réalisons nos éditions de code sur Atom et VisualStudioCode.

## Sommaire :

1. **Base de données :**
  - ➔ MCD et MPD
  - ➔ Création de la base de données
2. **Front-end :**
  - ➔ Premier aperçu du site sur papier
  - ➔ Page de connexion
  - ➔ Index/menu
  - ➔ Voir mes trajets
  - ➔ Chercher un trajet
  - ➔ Se déconnecter
  - ➔ Aperçu du code
3. **Back-end :**
  - ➔ Aperçu du JSON
  - ➔ Manipulation de la base de données den PHP
  - ➔ Page de redirection et session
4. **Techniques de débogages**
5. **Résultat Final et Conclusion**

## 1. Base de données : MCD et MPD :

Avant de démarrer la programmation et la base de données, nous avons décidé de passer en premier par une conceptualisation de la base de données, fondation du site. Nous avons donc décidé de développer sur un tableau notre MCD (Modèle Conceptuel de Données) dans lequel nous avons créé les premières tables de données nous semblant indispensables.

De là nous avons lié les différentes tables par des actions, et nous avons ensuite ajouté les différentes cardinalités correspondant aux tables et aux actions comme vu dans la ressource « source de données ».

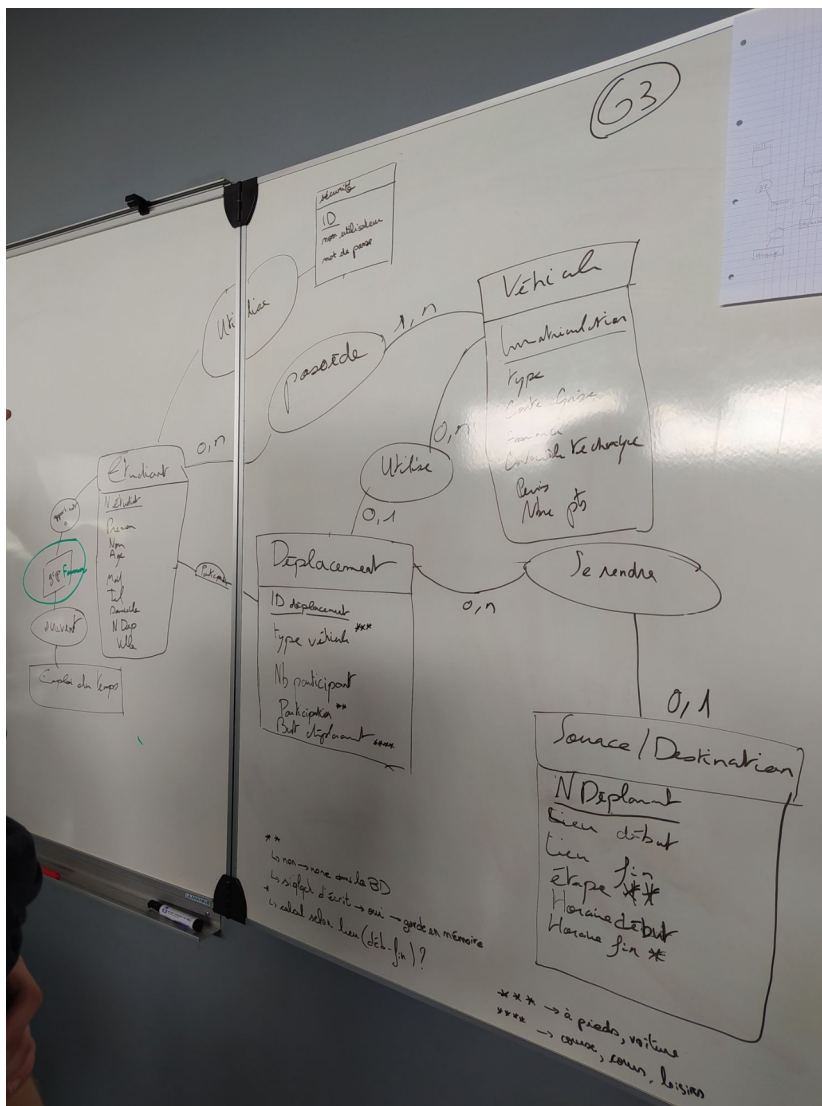


Figure 1 : Schéma du MCD

De là, depuis les cardinalités nous en avons déduit les tables à rajouter afin d'arriver sur une modélisation cohérente de la base de données. Nous avons donc fait un MPD (Modèle Physique de Données). Nous avons un modèle final comportant 9 tables. Nous avons ajouté les clés primaires et étrangères afin de lier ces tables entre elles.

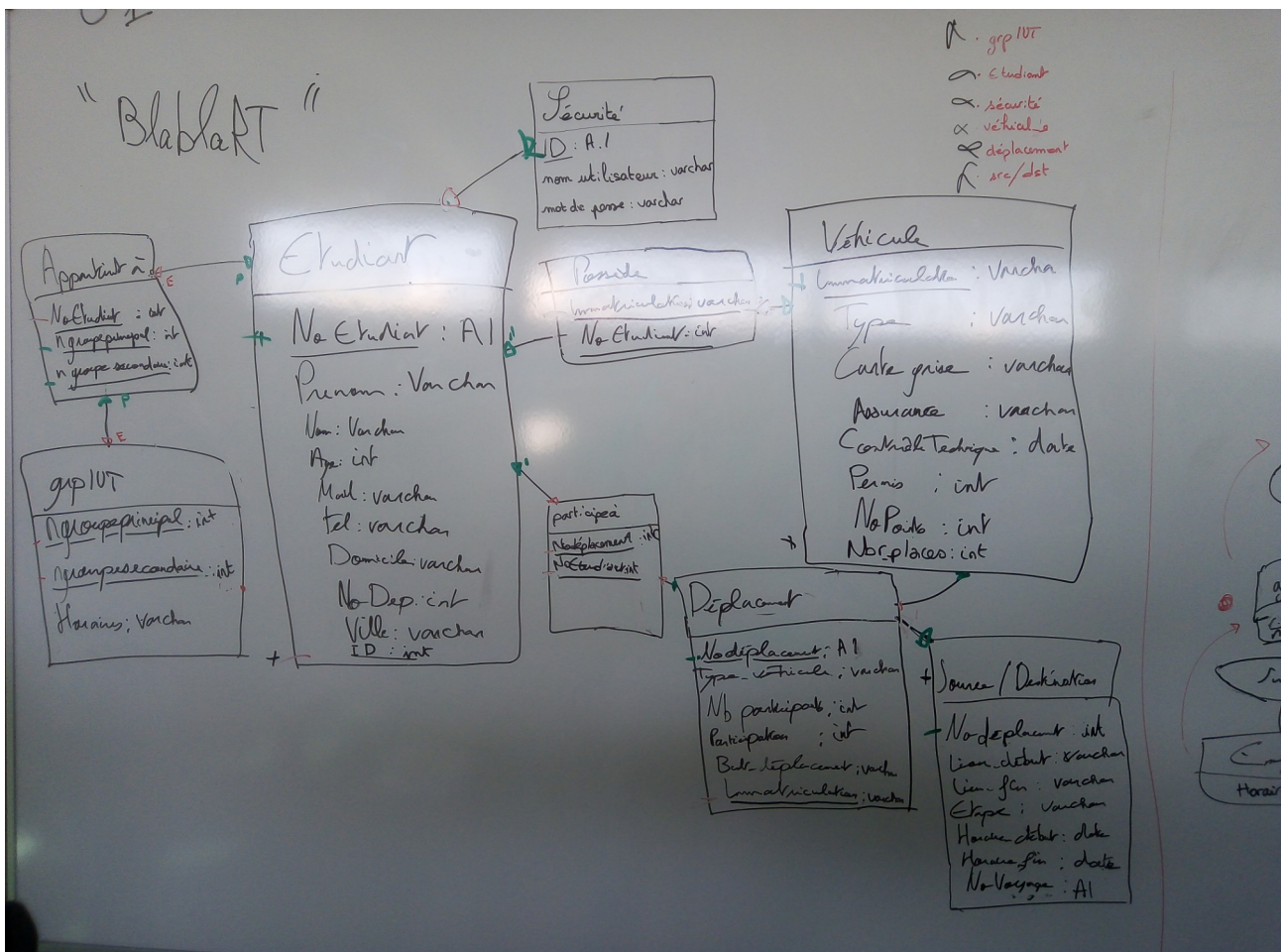


Figure 2 : Première version du MPD

Cependant, ce dernier étant trop complexe. Nous avons décidé sous conseil de nos professeurs de le simplifier ce qui nous a donné :

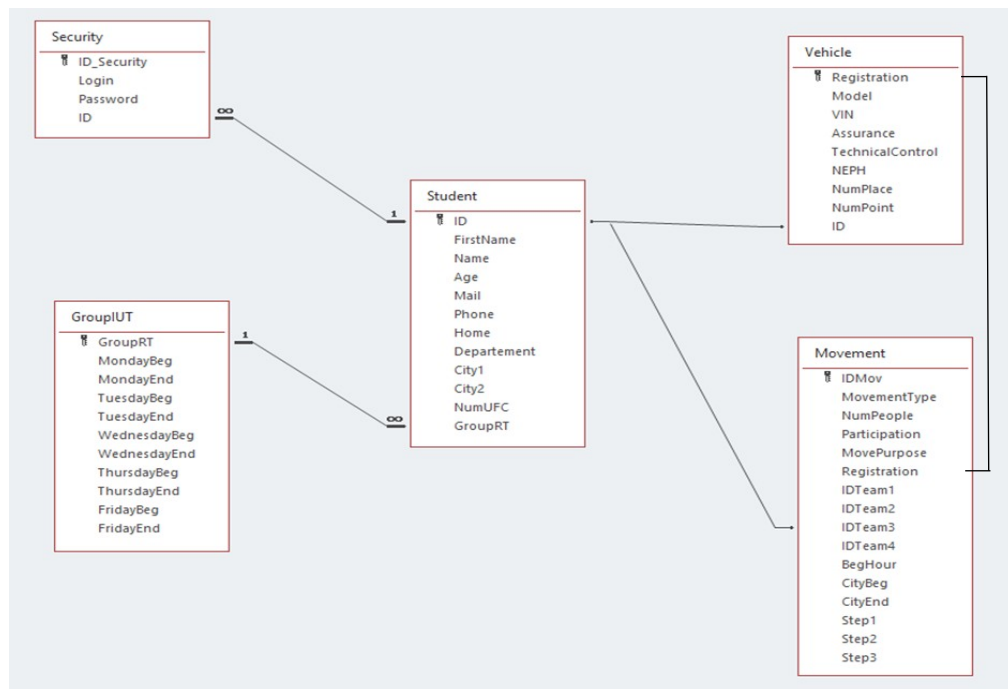


Figure 3 : version finale du MCD réalisé sous accès

### Création de la base de données :

Avant de réaliser la base de données sur PHPMyAdmin, nous avons réalisé une modélisation sur Access, afin d'avoir un aperçu du résultat définitif.

Une fois cette modélisation finie, nous avons pu passer sur la base en PhpMyAdmin. Nous avons décidé de la réaliser en anglais, ce qui, dans l'optique d'une éventuelle exportation à l'étranger rends la prise en main plus rapide. Avant d'avoir une base complète et cohérente nous nous y sommes pris à trois fois.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1	No_etudiant	int(100)		Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2	Prenom	varchar(100)	utf8_general_ci	Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3	Nom	varchar(100)	utf8_general_ci	Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4	Age	int(3)		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	5	Mail	varchar(100)	utf8_general_ci	Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	6	Tel	varchar(15)	utf8_general_ci	Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	7	Domicile	varchar(100)	utf8_general_ci	Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	8	Num_Dep	int(5)		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	9	Ville	varchar(100)	utf8_general_ci	Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	10	ID	int(100)		Non	Aucun(e)			Modifier Supprimer Plus

☐ Tout cocher    Avec la sélection :

1 colonne(s) après ID

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
<input type="button" value="Éditer"/> <input type="button" value="Supprimer"/>	PRIMARY	BTREE	Oui	Non	No_etudiant	0	A	Non	
<input type="button" value="Éditer"/> <input type="button" value="Supprimer"/>	ID	BTREE	Non	Non	ID	0	A	Non	

Créer un index sur 1 colonnes

Figure 4 : Table Student avec les types mis en avant

	ID	FirstName	Name	Age	Mail	Phone	Home	Department	City1	City2	NumUFC
<input type="checkbox"/> <input type="button" value="Éditer"/> <input type="button" value="Copier"/> <input type="button" value="Supprimer"/>	2	Thomas	Raynaud	19	traynau2@carnel.com	0606060606	10 rue de la piscine	25	Montbéliard	Rouffach	12345678
<input type="checkbox"/> <input type="button" value="Éditer"/> <input type="button" value="Copier"/> <input type="button" value="Supprimer"/>	3	Thomas	Mirbey	19	tmirbey@carnel.com	0707070707	10 rue des Vosges	25	Voujaucourt	None	98765432
<input type="checkbox"/> <input type="button" value="Éditer"/> <input type="button" value="Copier"/> <input type="button" value="Supprimer"/>	4	Zacharie	George	21	zgeorge2@carnel.com	123987654	10 rue du parc	25	Montbéliard	Colmar	78932146
<input type="checkbox"/> <input type="button" value="Éditer"/> <input type="button" value="Copier"/> <input type="button" value="Supprimer"/>	5	Yohann	Mittel	19	ymittel@gmail.com	6491537261	5 impasse du pommier	25	Bethoncourt	Guadeloupe	64315794
<input type="checkbox"/> <input type="button" value="Éditer"/> <input type="button" value="Copier"/> <input type="button" value="Supprimer"/>	6	Petit	Clément	18	clementpetit01@gmail.com	0635987565	9 rue du poirier	25	Montbéliard	Lons-le-saunier	78669518
<input type="checkbox"/> <input type="button" value="Éditer"/> <input type="button" value="Copier"/> <input type="button" value="Supprimer"/>	7	Amaud	Kastner	19	Amaudk@gmail.com	0794652917	3 impasse du chevreuil	25	Montbéliard	Strasbourg	76164995

Figure 5 : Remplissage de la table Student



## 2. Front-end :

### Premier aperçu du site sur papier :

Une fois la base de données terminée, le site pouvait être développé. Nous sommes au préalable passé par une visualisation papier du site, et un brainstorming. La première version n'a pas été retenue, mais une seconde version a été réalisée plus tard dans la semaine qui cette fois nous a été plus convaincante :

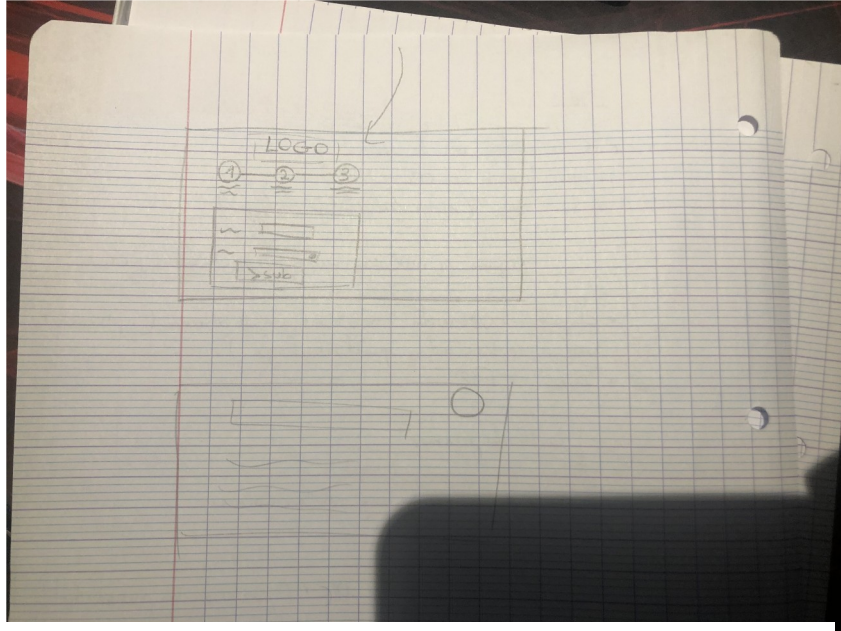


Figure 6 : Première version du visuel du site

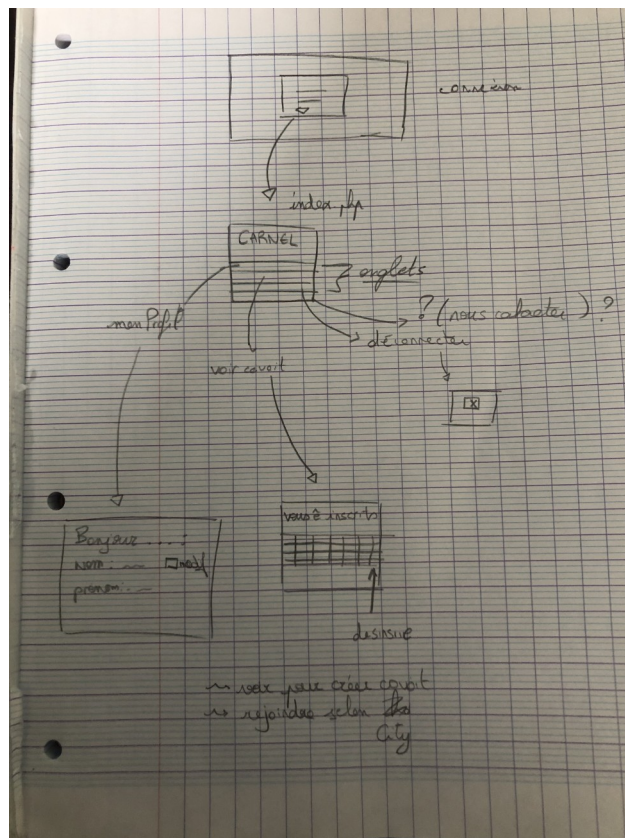
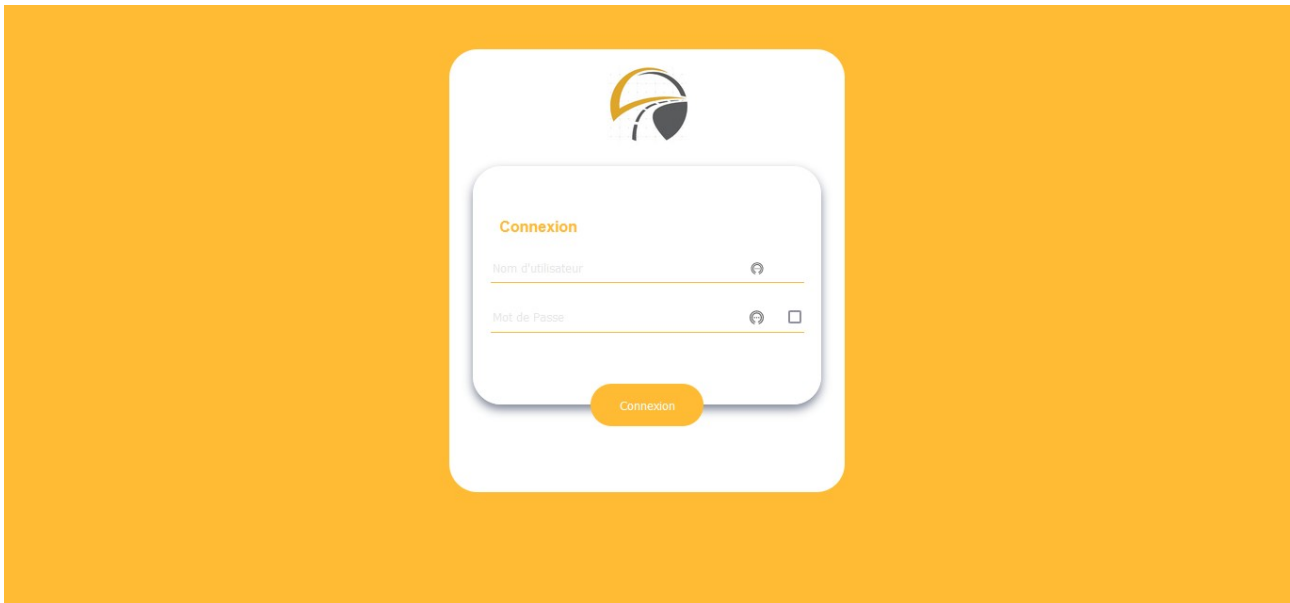


Figure 7 : Schéma final des interfaces du site

- ⇒ Il faut savoir que le front-end est la dernière chose que nous avons réalisé, en effet nous préférons avoir une application web fonctionnelle avant la fin de la semaine. La seule page que nous avons commencé auparavant est la page de connexion.

### Page de connexion :

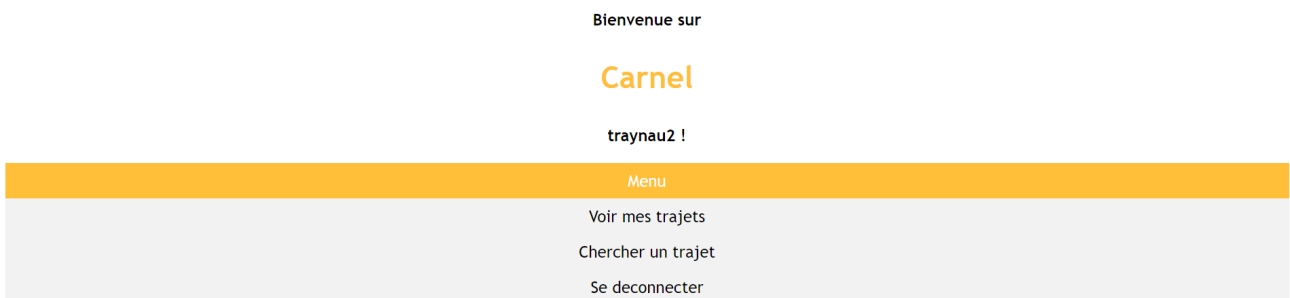


**Visuel et technologies utilisées :** sur cette page de connexion on trouve deux champs. L'un pour rentrer son identifiant, l'autre pour entrer son mot de passe. Mot de passe qui peut être visualisé grâce à un bouton à droite du champ réalisé en JavaScript. Cette page a entièrement été réalisée au niveau du front par de l'HTML et du CSS pour traiter de l'aspect visuel et du responsive en utilisant notamment @media qui permet d'adapter l'affichage en fonction de l'appareil utilisé. Et aussi du javascript comme expliqué aussitôt.

**Fonctionnement de la page :** une fois les bons identifiants et mots de passe rentrés, il ne reste plus qu'à cliquer sur le bouton « Connexion » pour se connecter, comme toute page de connexion classique. Les identifiants et mots de passe doivent être rentrés par un administrateur manuellement ou grâce à un fichier JSON que nous traiterons ultérieurement. Le bouton connexion renvoie sur l'index.php peu importe le groupe de l'utilisateur (étudiant ou administrateur) et que qu'un identifiant et mot de passe connu ont été rentrés.

### Index/Menu :

La page principale étant nommé index, est une sorte de menu. Dans ce menu, se trouve comme dans les pages que nous verrons ultérieurement des barres de navigation permettant de choisir dans quelle page aller.



Dans cette page se trouve aussi un court texte, présentant plusieurs choses : l'historique de cette application web, son rôle et son utilité et aussi comment obtenir un compte en tant qu'utilisateur.

## Carnel

### Histoire :

Carnel est une application web destinée à usage interne des étudiants travaillant dans l'IUT Réseaux et Télécommunications de Montbéliard. Elle a été développée dans le cadre de la Situation d'Apprentissage et d'Evaluation 23 par des étudiants en première année (Thomas MIRBEY, Thomas RAYNAUD et Zacharie GEORGE).

### Rôle et utilité :

Cette application permet d'organiser la mise en contact d'étudiants qui étudient dans la même ville, qui habitent dans la même ville et qui se rendent aux mêmes endroits pour ses études, ses loisirs ou pour ses courses. Cette application n'est pas seulement destinée au covoiturage, mais vous pouvez aussi y renseigner d'autres moyens de se déplacer comme par vélo, train, pieds ...

### Comment obtenir votre compte ?

Pour obtenir votre compte, il faut contacter un administrateur à l'adresse : [administration@carnel.com](mailto:administration@carnel.com).

Ces derniers peuvent vous inscrire sur la liste des participants avec le mot de passe que vous souhaitez et différentes informations que vous pouvez transmettre.

## Voir mes trajets

Cette page permet de visionner à quels covoiturages l'utilisateur connecté est inscrit la page est `show2.php`. Sous la forme d'un tableau avec plusieurs arguments. La dernière colonne contient un moyen pour se désinscrire d'un covoiturage.

Menu
Voir mes trajets
Chercher un trajet
Se deconnecter

Voici les covoiturages auxquels vous êtes inscrits :

Mode de déplacement	But du déplacement	Nombre de personne(s)	Participation	Personne 1	Personne 2	Personne 3	Personne 4	Se désinscrire
Voiture	Voiture	Cours	2	2\$	admin admin	Thomas Raynaud	X	X

Cette page est différente pour un compte administrateur la page est appelée `show_admin.php`. En effet, ce dernier peut visionner tous les covoiturages et les supprimer s'il le faut.

Menu
Voir mes trajets
Chercher un trajet
Se deconnecter

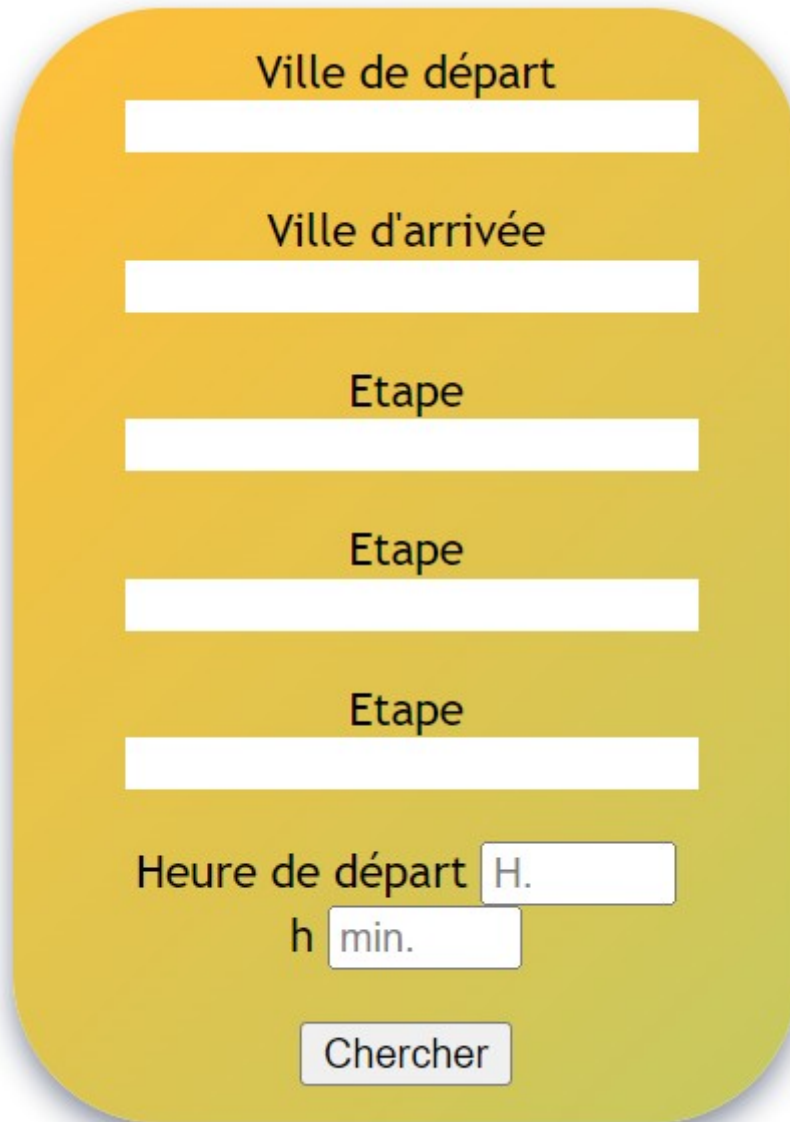
Vous êtes : admin. Vous avez donc des droits supplémentaires, visionner tous les covoiturages ainsi que de les supprimer.

Type de transport	Type de participation	Ville de départ	Ville d'arrivée	Heure de départ	Participant1	Participant2	Participant3	Participant4	Supprimer
Voiture	2\$	Rouffach	Montébliard	12h30	admin admin	Thomas Raynaud	X	X	<input type="checkbox"/>
Voiture	2\$	Rouffach	Montébliard	12h30	X	X	X	X	<input type="checkbox"/>

Cette page a été réalisée uniquement pour le front à l'aide d'HTML et de CSS pour le visuel bien que le PHP facilite la création et le remplissage de tableau avec des boucles.

### **Chercher un trajet**

Afin qu'un utilisateur puisse visionner tous les covoiturages et aussi ajuster sa recherche avec des filtres, nous avons réalisé cette page. Pour affiner la recherche, des filtres avec la ville de départ, ville d'arrivée, les différentes étapes et l'heure de départ.



Form fields and labels inside the yellow container:

- Ville de départ
- Ville d'arrivée
- Etape
- Etape
- Etape
- Heure de départ H. min.
- Chercher

*Figure 8 : Formulaire permettant le filtrage des covoiturage*



Les trajets sont ensuite présentés comme ceci :

## Trajet numéro 1

- Ville de départ : Rouffach
- Destination : Montébliard
- Participation : 2\$
- Moyen de transport : Cours

## Trajet numéro 2

- Ville de départ : Rouffach
- Destination : Montébliard
- Participation : 2\$
- Moyen de transport : Cours

Il faut quand même savoir qu'initialement ils sont tous présentés à l'utilisateur.  
Cette page utilise exactement les mêmes technologies que la page précédente.

### **Se déconnecter**

On peut aussi voir un bouton permettant de fermer la session de l'utilisateur actif à un moment donné.

Se deconnecter

## Aperçu du code :

Voici le code permettant de montrer le mot de passe à l'utilisateur dans la page de connexion en javascript :

```
85     <script>
86         //Ce script permet d'afficher le mot de passe de l'utilisateur
87         var txt=document.getElementById('MDP');
88
89         // On ajoute un evenement qui s'excutera quand on cliquera
90         // sur le bouton et qui executera la fonction updateBtn
91         btn.addEventListener('click', updateBtn);
92
93         // Fonction qui d'afficher le mot de passe
94         function Afficher() {
95             //On récupère le champ dans lequel on tape le mot de passe
96             var input = document.getElementById("MDP");
97             //Si le mot de passe n'est pas affiché, on change la valeur password
98             //En text pour pouvoir le voir
99             if (input.type === "password")
100             {
101                 input.type = "text";
102             }else{
103                 //Sinon on passe la valeur en password pour ne plus l'afficher
104                 input.type = "password";
105             }
106         }
107     </script>
```

La structure du menu de navigation peut aussi être présenté (HTML) :

```
33     <tr>
34         <td>
35             <ul class="test">
36                 <li><a class="active" href="index.php">Menu</a></li>
37                 <li><a href="redirect.php">Voir mes trajets</a></li>
38                 <li><a href="covoiturage_modif.php">Chercher un trajet</a></li>
39                 <li><a href="deconnexion.php">Se deconnecter</a></li>
40             </ul>
41         </td>
42     </tr>
```

On peut voir la classe appelée « active » qui permet de voir dans quel page on est actuellement, ce qui donne en CSS :

```
.active {  
    background-color: #FFBB33;  
    color: white;  
}
```

➔ On peut voir le background en orange et l'écriture en blanc.

### 3. Back-End :

#### Aperçu du JSON :

Une des difficultés imposées, et que nous avons rencontré lors de la réalisation de ce projet est le remplissage de la base de données à l'aide d'un fichier JSON.

La création du fichier JSON a été plutôt simple à l'aide du Wikipédia JSON.

```
"0": [  
  {  
    "Informations":  
    {  
      "FirstName": "Thomas",  
      "Name": "Raynaud",  
      "Age": "19",  
      "Mail": "traynau2@carnel.com",  
      "Phone": "0781230626",  
      "NumUFC": "142537",  
      "GroupRT": "RT1-LK2",  
      "Adresses":  
      {  
        "Home": "5 rue de la piscine",  
        "Department": "68250",  
        "City1": "Rouffach",  
        "City2": "Montbéliard"  
      }  
    }  
  },  
  ]
```

Cependant nous devons créer un script permettant d'exécuter et de remplir automatiquement la base de données à l'aide du fichier JSON.

Dans un premier temps, nous avons commencé à réaliser le script en php, mais devant la difficulté et la longueur de ce dernier, nous avons lors décider de le réaliser en Python, un langage bien maîtrisé par un membre du groupe. Malgré que ce dernier fonctionnait en local sur un serveur

XAMPP, il ne fonctionnait pas sur le serveur rt-serv à cause du manque d'un paquet permettant l'utilisation de requêtes SQL en python.

```
61 assurance=data[2]["Vehicule"]['Assurance']
62 controletechnique=data[2]["Vehicule"]['ControleTechnique']
63 NEPH=data[2]["Vehicule"]['NEPH']
64 noplaces=data[2]["Vehicule"]['NoPlaces']
65 nopoints=data[2]["Vehicule"]['NoPoints']
66
67 #-----
68
69 typedeplacement=data[3]["Deplacement"]['TypeDeplacement']
70 nbrparticipants=data[3]["Deplacement"]['NbrParticipants']
71 participation=data[3]["Deplacement"]['Participation']
72 butdepacement=data[3]["Deplacement"]['ButDeplacement']
73
```

Nous avons donc dû revenir sur du PHP. Après plusieurs heures de développement, nous avons obtenu un script fonctionnel. Dans ce dernier il faut respecter plusieurs règles, comme commencer la déclaration d'un utilisateur par 0, puis 1 ... Il faut aussi avoir les mêmes noms de variables que dans notre base de données.

```
foreach ($array as $keyStart => $valueStart) { //tous Les foreach utilisés permettent de rentrer dans Le
    if ($keyStart==strval($n)) { //si on est bien dans un utilisateur dans Le json (0,1...)
        foreach ($valueStart as $key => $value) {
            foreach ($value as $keyI => $valueI) { //on rentre dans Le tableau et on obtient Les information
                if($keyI=='Informations'){//si la clé trouvé c'est information alors on fait Le traitement
                    foreach ($valueI as $keyA => $valueA) { //on va voir ce qu'il y a dans information
                        if($keyA!='Adresses'){//si tout sauf adresse
                            $informations .= "'$valueA' . " , "; //on met à la suite de la variable la valeur trou
                            $informationskey .= $keyA." , "; //pareil mais avec la clé de cette valeur
                            $information2.=$keyA."="."'$valueA'." AND "; //dans la variable, on ajoute à la suite
                        } else { //dans l'autre cas ou la clé est "Adresses"
                            foreach ($valueA as $keyO => $valueO) { //on a un sous tableau dans ce cas
                                $informations .= "'$valueO' . " , "; //on refait Le même traitement que ligne 34-35
                                $informationskey .= $keyO." , ";
                            }
                        }
                    }
                }
            }
        }
    }
}
```

### Manipulation de la base de données en PHP :

Afin de lier nos pages web à la base de données, il fallait alors réaliser des requêtes SQL en PHP.

Nous avons dans ce cas réaliser des requêtes INSERT :

```
$sql4= "INSERT INTO Movement ($deplacementkey, Registration) VALUES ($deplacement,'$registration');";
```

Dans un second cas, ce qui nous a pris du temps était de lier les clés primaires avec les clés étrangère, dans ce cas on fait des requêtes SELECT comme utilisées dans l'exécutable JSON (json.php) :

```
$sql2bis="(SELECT ID FROM Student WHERE $information2)";
```

Nous avons aussi réalisé des requêtes UPDATE afin de supprimer des données comme dans delete.php :

```
$sql = "UPDATE movement SET IDTeam1 = $p1,IDTeam2 = $p2,IDTeam3 = $p3,IDTeam4 = $p4 WHERE IDmov=\"$var\"";
```

## Page de redirection et session :

Nous avons réalisé une page de redirection, ce dernier contrôle à l'aide de requêtes en PHP le numéro de l'utilisateur afin de vérifier si la personne connectée est un administrateur ou non. Selon le résultat, la page renvoyée sera alors différente. La page se nomme : redirect.php.

```
if($id==1){
    echo "1";
    $redirect_page = 'show_admin.php';
    header('Location: ' . $redirect_page);
}else{
    //echo"!1";
    $redirect_page = 'show2.php';
    header('Location: ' . $redirect_page);
}
```

On peut voir que si l'ID est de 1, on aura affaire à un administrateur, tandis que dans les autres cas se sera des utilisateurs lambda. La redirection sera alors différente entre show\_admin.php et show2.php.

Nous avons aussi utilisé la fonction session\_start() en php, qui se déclare au début du code. Cette dernière permet de commencer une session ou de continuer celle existant.

```
session_start();
```

Afin de fermer la session, on utilise dans la page déconnexion la fonction créée « unlog » contenant 2 autres fonctions :

```
function unlog(){
    session_destroy(); }
unlog();
?>
```

### 4. Techniques de débogage

Afin de corriger nos erreurs lors du code, nous avons dû utiliser des techniques de débogage. La première et la plus utilisée nous a servi afin de vérifier la bonne écriture de nos requêtes sql. En effet en général, nos requêtes se trouvent dans une variable tel que \$sql. Nous réalisons alors des **echo \$sql** afin de voir ce que contenait notre requête.

```
$sql1= "INSERT INTO Student ($informationskey) VALUES ($informations)";//re
//echo "<br>" . "<br>".$sql1;
```

Figure 9 : Utilisation d'echo afin de voir le contenu de la requête sql en php

Nous avons aussi utilisé une autre manière plus basique cette fois. Nous avons utilisé les codes d'erreurs afin d'améliorer notre code et obtenir un code plus correct.



## Conclusion :

Bien que manquant d'un côté esthétique certain lors de la présentation du vendredi 13 mai 2022, tous les éléments **nécessaires** au fonctionnement de l'application sont présents. On y voit notamment une bonne perspective d'évolution au cours du temps à l'aide d'une base de données évolutive. Certains éléments demandés ne sont quand même pas mis en place dans notre application web, comme le fait de pouvoir créer un covoiturage ou d'un rejoindre, d'ajouter des véhicules. Les étudiants de l'IUT peuvent se déplacer ensemble via l'application en réservant des trajets ou des déplacements en commun. Les différents moyens de transport peuvent être choisis, et l'on peut également choisir la cause du déplacement comme les études, les courses, ou les loisirs. La création de groupe est aussi utile quant à l'utilisation des emplois du temps que nous n'avons pas eu le temps de développer.

Afin de mettre en production notre projet, nous avons utilisé github notamment pour le partage au lien suivant : <https://github.com/ThomasM2568/Sae23-George-Raynaud-Mirbey> .

En conclusion, ce projet nous a appris à mener à bien un projet en équipe, va nous permettre de mieux gérer le temps mis à notre disposition pour nos prochains projets, mais aussi à développer une familiarité avec la gestion de projet et les outils permettant une organisation optimisée dès le début du projet, afin d'éviter les pertes de temps.

Les pistes d'amélioration sont bien entendu le rendu esthétique du site, même si nous nous sommes beaucoup plus concentrés sur l'aspect technique et fonctionnel de l'application web. On peut aussi développer certains points manquants dans le cahier des charges qui nous auraient intéressés si nous avions eu plus de temps et que nous allons éventuellement continuer dans le futur afin de nous améliorer encore plus.

Pour toute question relative à notre travail, nous sommes joignable à :

- [thomas.mirbey@edu.univ-fcomte.fr](mailto:thomas.mirbey@edu.univ-fcomte.fr)
- [thomas.raynaud@edu.univ-fcomte.fr](mailto:thomas.raynaud@edu.univ-fcomte.fr)
- [zacharie.george@edu.univ-fcomte.fr](mailto:zacharie.george@edu.univ-fcomte.fr)

Crédit : Thomas MIRBEY, Thomas RAYNAUD et Zacharie GEORGE.

Professeurs encadrants : Monsieur DEDU et Monsieur CANALDA.