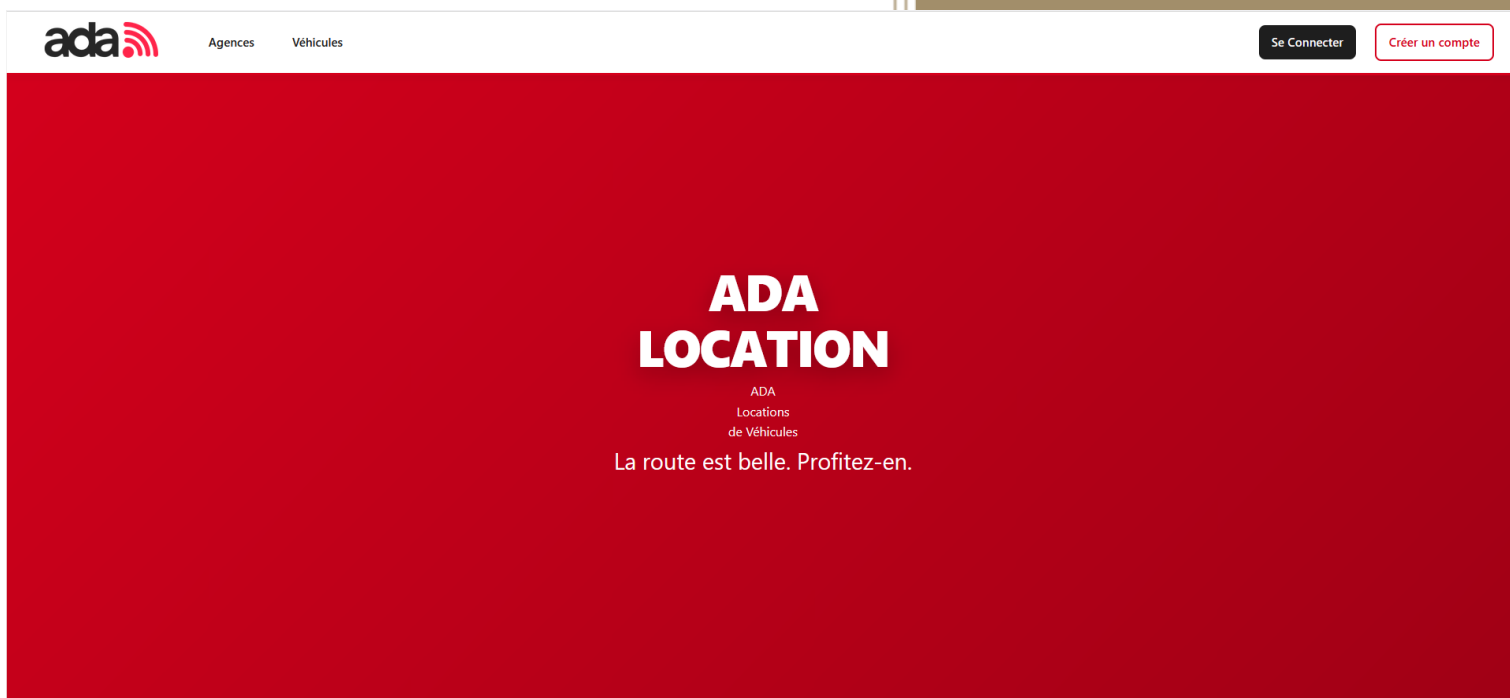




2025-
2026

Projet ADA-Phase2 : Laravel



Sommaire

I.	Expressions des besoins	3
a.	Contexte	3
b.	Spécifications fonctionnelles	3
c.	Base de données	4
1.	Schema Relationnel	4
2.	Diagramme de classes UML	5
II.	Fonctionnalités.....	5
a.	Fonctionnalité 1 : Informations agences	5
b.	Fonctionnalité 2 : Informations véhicules	7
c.	Fonctionnalité 3 : Création compte client	8
d.	Fonctionnalité 4 : Connexion/Déconnexion.....	10
e.	Fonctionnalité 5 : Réservation location	11
f.	Fonctionnalité 6 : Visualisation des contrats	14
g.	Routes	15
h.	Controller.....	15
III.	Plannification du projet	16

I. Expressions des besoins

a. Contexte

Vous faites partie d'une équipe Scrum. Votre mission consiste à participer au développement du site web de ADA. En mode agile, vous avez la charge d'analyser les spécifications techniques pour concevoir et développer l'application frontend/backend permettant à tout internaute de consulter les données de ADA. Cette application accèdera à une base de données Mysql centralisée commune à toutes les agences. Vous testez et intégrez en continu la solution développée.

L'application sera accompagnée d'une documentation écrite et structurée relative à la gestion du projet selon la documentation jointe. La gestion du projet se fera en mettant en œuvre un outil de gestion de projet (Trello, Projector) et éventuellement un outil de gestion de développement et de versions (Github).

b. Spécifications fonctionnelles

Les fonctionnalités réalisées sont :

- Fonctionnalité 1 : En tant que visiteur du site, je veux sélectionner dans une liste déroulante une agence pour en visualiser les informations complètes : adresse, téléphone, email, les horaires d'ouverture, et tous les véhicules de l'agence (marque, modèle, libellé et photo catégorie).
- Fonctionnalité 2 : En tant que visiteur du site, je veux visualiser toute l'offre de location de ADA, c'est-à-dire que je veux sélectionner dans une liste déroulante (ou icônes cliquables) un genre, puis dans une deuxième liste déroulante des catégories correspondantes (ou icônes cliquables), je veux sélectionner une catégorie pour en visualiser les informations complètes : libelle, ..., photo, et les attributs (logo, libellé, et valeur) de la catégorie.

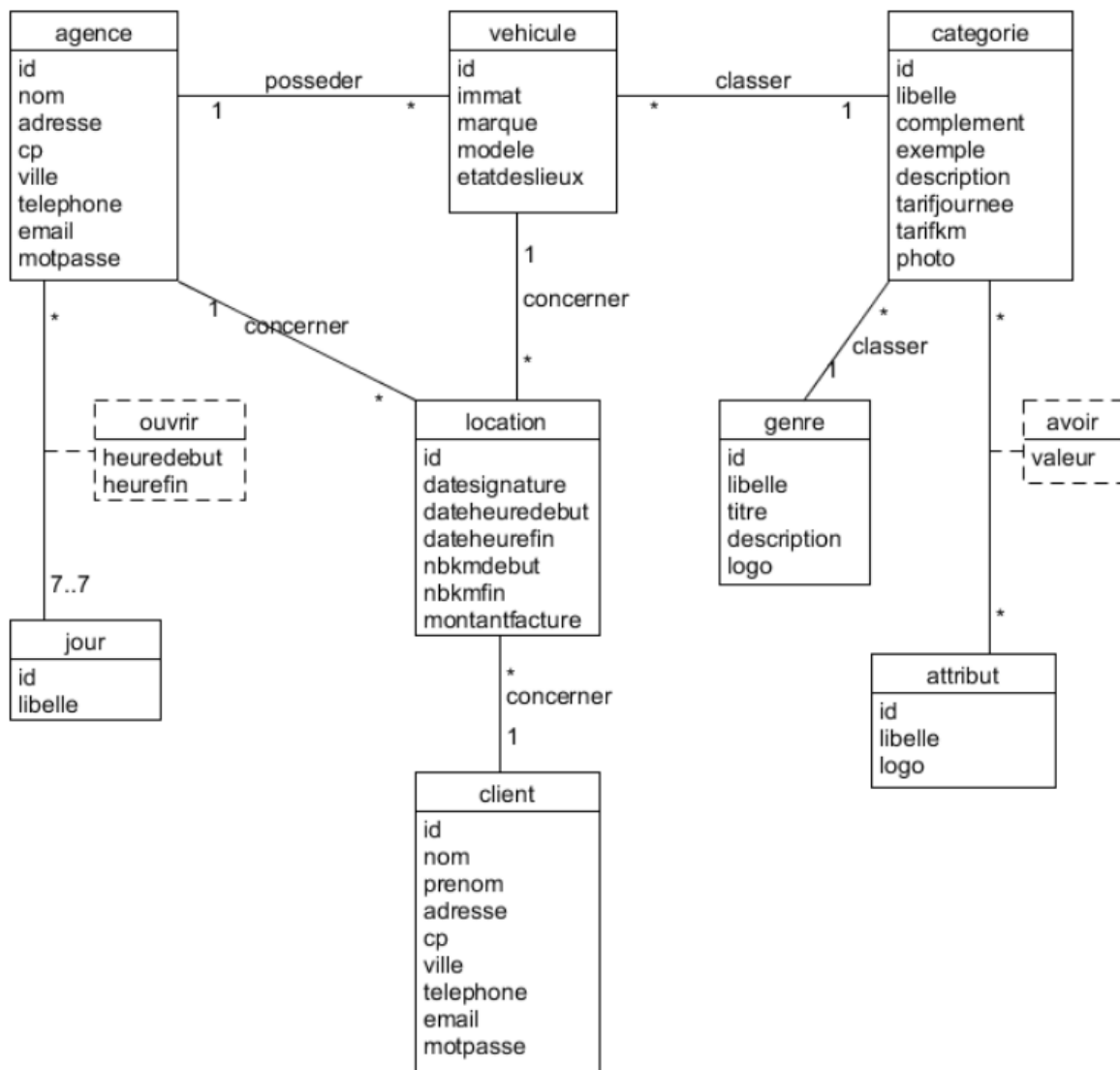
- Fonctionnalité 3 : En tant que visiteur du site, je veux me créer un compte client.
- Fonctionnalité 4 : En tant que visiteur du site, je veux me connecter avec mon compte client et me déconnecter. Mon menu s'adapte en fonction de ma situation (connecté ou non).
- Fonctionnalité 5 : En tant que client connecté, je veux réserver une location, c'est-à-dire que je veux saisir une date de début et de fin de location ainsi que sélectionner dans une liste déroulante une agence, pour en visualiser tous les véhicules disponibles (qui ne sont pas déjà réservés sur la même période). Puis je veux pouvoir sélectionner un véhicule pour pouvoir enregistrer ma réservation de location.
- Fonctionnalité 6 : En tant que client connecté, je veux visualiser tous mes contrats de location (passés ou en cours), puis pouvoir cliquer sur un contrat pour en visualiser les informations détaillées : dates, nbkm, montant, agence, véhicule.

c. Base de données

1. Schema Relationnel

1	agence (<u>id</u> , nom, adresse, cp, ville, telephone, email, motpasse)
2	vehicule (<u>id</u> , immat, marque, modele, etatdeslieux, categorie_id, agence_id)
3	categorie (<u>id</u> , libelle, complement, exemple, description, tarifjournee, tarifkm, photo, genre_id)
4	genre (<u>id</u> , libelle, titre, description, logo)
5	attribut (<u>id</u> , libelle, logo)
6	location (<u>id</u> , datesignature, dateheuredebut, dateheurefin, nbkmdebut, nbkmfin, montantfacture, vehicule_id, agence_id, client_id)
7	client (<u>id</u> , nom, prenom, adresse, cp, ville, telephone, email, motpasse)
8	jour (<u>id</u> , libelle)
9	agence_jour (<u>agence_id</u> , <u>jour_id</u> , heuredebut, heurefin)
10	categorie_attribut (<u>categorie_id</u> , <u>attribut_id</u> , valeur)

2. Diagramme de classes UML



II. Fonctionnalités

a. Fonctionnalité 1 : Informations agences

Voici l’affichage pour les informations de l’agence. Les informations sont affichées après avoir choisi une agence.



Agence

ADA Mâcon
 4 Quai Jean Jaurès 71000 Mâcon
 Téléphone : 0385204000
 Email : macon@ada.fr

Horaires

Lundi	08:00 - 19:00	Mardi	08:00 - 19:00
Mercredi	08:30 - 19:30	Jeudi	08:00 - 19:00
Vendredi	08:00 - 19:00	Samedi	08:30 - 18:00
Dimanche	00:00 - 00:00		

Voitures de l'Agence

Véhicule	Catégorie	Photo
Peugeot 208	Fun Confort	
Renault Twingo	ADA Premier Prix	

Pour le code, il se présente comme ceci :

```


12 // Génération dynamique des options d'agences
13 foreach($lesagences as $uneligne) {
14     echo "<option value=\"" . $uneligne->id . "\"" . $uneligne->nom . "</option>";
15 }
16 ?>
17 </select>
18 <input type="submit" value="Valider">
19 </form>
20
21 <?php if(isset($uneAg)) { ?>
22 <!-- Affichage détaillé de l'agence sélectionnée -->
23 <div class="agence-info">
24     <h2>Agence</h2>
25     <div class="agence-details">
26         <p><strong><?php echo $uneAg->nom; ?></strong></p>
27         <p><?php echo $uneAg->adresse . " " . $uneAg->cp . " " . $uneAg->ville; ?></p>
28         <p>Téléphone : <?php echo $uneAg->telephone; ?></p>
29         <p>Email : <a href="mailto:<?php echo $uneAg->email; ?>"><?php echo $uneAg->email; ?></a></p>
30     </div>

```

Ce code permet de générer les agences dans un combobox et va permettre de sélectionner une agence. Une fois sélectionner et le bouton valider appuyer, Toutes les informations de l'agence : adresse, horaire et véhicule disponible seront affichés.

b. Fonctionnalité 2 : Informations véhicules








L'affichage de des véhicules se montre comme ceci :



ADA Premier Prix

Exemple : Ford Ka ou similaire
Tarif : 29€/jour + 0.2€/km

Attributs

Icône	Attribut	Valeur
	Bagages	1 bagage
	Energie	Electrique ou Essence
	Places	4 places
	Portes	3 portes
	Sacs	1 sac
	Age	+21 ans
	Permis	permis 1 an

Cette page montrera les différentes informations du véhicule sélectionné.

La page a été configurée comme ceci :

```
41 <!-- Détails de la catégorie sélectionnée -->
42 <div class="categorie-content">
43 <div class="categorie-header">
44 <div class="categorie-image">
45 libelle; ?>">
46 </div>
47 <div class="categorie-info">
48 <h2><?php echo $lacategorie->libelle; ?></h2>
49 <p><strong>Exemple :</strong> <?php echo $lacategorie->exemple; ?></p>
50 <p><strong>Tarif :</strong> <?php echo $lacategorie->tarifjournee; ?>€/jour + <?php echo $lacategorie->tarifkm; ?>€/km</p>
51 </div>
52 </div>
53
54 <!-- Tableau des attributs via la table relationnelle d'attribut_categorie -->
55 <h3>Attributs</h3>
56 <table class="attributs-table">
57 <thead>
58 <tr>
59 <th>Icône</th>
60 <th>Attribut</th>
61 <th>Valeur</th>
62 </tr>
63 </thead>
```

Ce code va permettre d'afficher les différentes informations du véhicule dans un tableau. Le véhicule a été choisi suivant sa catégorie.

c. Fonctionnalité 3 : Création compte client

La page de la création du compte client se présente comme ceci :

CRÉATION D'UN COMPTE CLIENT

Informations de connexion

Adresse mail :

Mot de passe (minimum 6 caractères) :

Informations personnelles

Nom :

Prénom :

Adresse :

Code postal :

Ville :

Téléphone :

CRÉER MON COMPTE

Déjà un compte ? [Se connecter](#)

Pour se créer un compte client, l'utilisateur va devoir rentrer plusieurs informations dont un email et un mot de passe à se rappeler pour se connecter. Il devra aussi enregistrer d'autres informations (nom, prénom, adresse, code postal, ville et téléphone) pour se créer un compte client.

```

21 <h3>Informations de connexion</h3>
22 <label>Adresse mail :</label>
23 <input type="email" name="txtMail" required><br>
24
25 <label>Mot de passe (minimum 6 caractères) :</label>
26 <input type="password" name="txtMdp" minlength="6" required><br><br>
27
28 <h3>Informations personnelles</h3>
29 <label>Nom :</label>
30 <input type="text" name="txtNom" required><br>
31
32 <label>Prénom :</label>
33 <input type="text" name="txtPrenom" required><br>
34
35 <label>Adresse :</label>
36 <input type="text" name="txtAdresse" required><br>
37
38 <label>Code postal :</label>
39 <input type="text" name="txtCp" required><br>
40

```

On rentre les informations du client dans les labels puis les informations s'enregistre dans la base de données.

d. Fonctionnalité 4 : Connexion/Déconnexion

Si on est déconnecté la page affichera le bouton « Se connecter » :

Se Connecter

Créer un compte

Et si on est connecté alors la page affichera le bouton « Se déconnecter » et le nom et prénom du client :

VILOTTE KYLLIAN

Se Déconnecter

Pour se connecter, l'utilisateur va rentrer son adresse mail et son mot de passe et après que les informations soient vérifiées alors la connexion sera faite sinon un message d'erreur est envoyé et les informations doivent être vérifiées et réécrites.

```

15 <h1>Connexion</h1>
16 <div class="login-container">
17   <!-- Formulaire d'authentification -->
18   <form action="connexion" method="POST">
19     <input type="hidden" name="_token" value="<?php echo csrf_token() ?>" />
20
21     <label>Adresse mail :</label>
22     <input type="email" name="txtmail" value="<?php echo old('txtmail', ''); ?>" required<br>
23
24     <label>Mot de passe :</label>
25     <input type="password" name="txtmotpasse" required<br>
26
27     <input type="submit" value="Se connecter">
28
29     <div class="form-links">
30       <!-- Lien vers création de compte -->
31       <p>Pas encore de compte ? <a href="<?php echo url('/client'); ?>">Créer un compte</a></p>
32     </div>
33   </form>

```

e. Fonctionnalité 5 : Réservation location

Le client peut réserver un véhicule sur une période et une agence et le client pourra choisir le véhicule qu'il veut.

RÉSERVATION D'UN VÉHICULE

Sélectionner une date et heure de début de location :

16 / 12 / 2025 15 : 00
📅

Sélectionner une date et heure de fin de location :

17 / 12 / 2025 10 : 00
📅

Sélectionner une agence :

ADA Chalon
▼

RECHERCHER LES VÉHICULES

Le code de la page ressemble à ceci :

```

29 <!-- Formulaire de réservation avec véhicules disponibles -->
30 <form action="reservation" method="POST">
31 <!-- Transmission des paramètres en hidden -->
32 <input type="hidden" name="dateheuredebut" value="<?= $deb->format('Y-m-d H:i:s') ?>">
33 <input type="hidden" name="dateheurefin" value="<?= $fin->format('Y-m-d H:i:s') ?>">
34 <input type="hidden" name="agence_id" value="<?= $uneAg->id ?>">
35 <input type="hidden" name="_token" value="<?= csrf_token() ?>">
36
37 <h2>Véhicules disponibles</h2>
38 <p>Du <?= $deb->format("d/m/Y H:i") ?> au <?= $fin->format("d/m/Y H:i") ?></p>
39 <p>Sélectionner un véhicule :</p>
40

```

Cette page va demander une date, une heure et une fois qu'un véhicule est disponible de n'importe qu'elle agence alors le véhicule avec sa catégorie sera affiché comme ceci :

Véhicules disponibles

Du 16/12/2025 15:00 au 17/12/2025 10:00

Sélectionner un véhicule :



Peugeot 108

ADA Premier Prix



Mini Countryman

Fun Premium



Après avoir choisi un véhicule, le client va pouvoir appuyer sur le bouton « Enregistrer la réservation » va ouvrir une nouvelle page est affiché un message pour valider la réservation :

ENREGISTREMENT LOCATION

Votre réservation a bien été enregistrée

Numéro de location : 4

Date de signature : 19/12/2025

Période : Du 16/12/2025 15:00 au 17/12/2025 10:00

Statut : Confirmée

Vous pouvez consulter cette location dans l'onglet **Locations**

[Retour à l'accueil](#)

Le code de la page représenter ainsi :

```
11 <!-- Détails de la location créée -->
12 <div class="confirmation-details">
13   <ul>
14     <li>
15       <span class="confirmation-label">Numéro de location :</span>
16       <span class="confirmation-value"><?= $location->id ?></span>
17     </li>
18     <li>
19       <span class="confirmation-label">Date de signature :</span>
20       <span class="confirmation-value"><?= $location->datesignature->format('d/m/Y') ?></span>
21     </li>
22     <li>
23       <span class="confirmation-label">Période :</span>
24       <span class="confirmation-value">
25         Du <?= $location->dateheuredebut->format('d/m/Y H:i') ?>
26         au <?= $location->dateheurefin->format('d/m/Y H:i') ?>
27       </span>
28     </li>
29   </ul>
```

Il va permettre d'afficher les informations de la réservation du véhicule.

f. Fonctionnalité 6 : Visualisation des contrats

Le client pourra visualiser toutes les réservations de véhicules qu'il a fait :

LOCATIONS

1

Location n°4 du 19/12/25

Début : 16/12/25 15:00


Fin : 17/12/25 10:00

Montant total : Non calculé €

Agence : ADA Chalon - 385 Boulevard
Victor Hugo 71100 Chalon sur Saône

Véhicule : LM-450-PM -
Citroën

Catégorie : Super Eco



La page de code a été structuré comme ceci pour que les locations soient affichées dans des cadres avec plusieurs informations.

```
15     foreach($lesLocations as $uneL) {
16         echo '<li>';
17         echo '<div class="location-badge">' . $loop . '</div>';
18         echo '<h3>Location n°' . $uneL->id . ' du ' . $uneL->datesignature->format('d/m/y') . '</h3>';
19         echo '<p><strong>Début :</strong> ' . $uneL->dateheuredebut->format('d/m/y H:i') . '</p>';
20         echo '<p><strong>Fin :</strong> ' . $uneL->dateheurefin->format('d/m/y H:i') . '</p>';
21         echo '<p><strong>Montant total :</strong> ' . ($uneL->montantfacture ?? 'Non calculé') . ' €</p>';
22         // Navigation dans les relations Eloquent
23         echo '<p><strong>Agence :</strong> ' . $uneL->agence->nom . ' - ' . $uneL->agence->adresse . ' ' . $uneL->agence->cp . ' ' . $uneL->agence->ville . '</p>';
24         echo '<div class="location-vehicule">';
25         echo '<div>';
26         echo '<p><strong>Véhicule :</strong> ' . $uneL->vehicule->immat . ' - ' . $uneL->vehicule->marque . '</p>';
27         echo '<p><strong>Catégorie :</strong> ' . $uneL->vehicule->category->libelle . '</p>';
28         echo '</div>';
29         echo 'vehicule->category->libelle . ' ' . $uneL->vehicule->immat . '>';
30         echo '</div>';
31         echo '</li>';
```

g. Routes

Pour relier chaque page entre elles, il fallait créer des routes comme ceci :

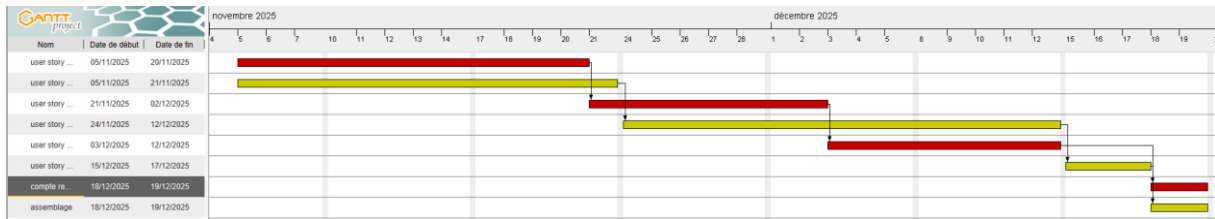
```
6 Route::get('/', [AdaC::class, 'index'])->name('accueil');
7
8 // User 1 - Consultation
9 Route::get('/agences', [AdaC::class, 'agences']);
10
11 // User 2 - Catalogue
12 Route::get('/vehicule', [AdaC::class, 'vehicule'])->name('vehicule');
13
14 // User 3 - Inscription
15 Route::get('/client', [AdaC::class, 'clientSaisir'])->name('client');
16 Route::post('/ajoutersave', [AdaC::class, 'clientCreer']);
17
18 //User 4 - Authentification (Login/Logout)
19 Route::get('/connexion', [AdaC::class, 'connexionSaisir']);
20 Route::post('/connexion', [AdaC::class, 'connexionExecuter']);
21 Route::get('/deconnexion', [AdaC::class, 'deconnexion']);
22
23 //User 5 - Processus de Réservation (Recherche puis Validation)
24 Route::get('/reservation', [AdaC::class, 'locationSaisir'])->name('reservation');
25 Route::post('/reservation', [AdaC::class, 'locationCreer']);
26
27 //User 6 - Espace Client
28 Route::get('/locations', [AdaC::class, 'locations']);
```

h. Controller

Pour relier des pages entre elles, il fallait aussi créer des fonctions dans un controller pour savoir ce que la page devait faire. En voici un exemple :

```
98 public function clientCreer(Request $request)
99 {
100     // Validation des données entrantes
101     $request->validate([
102         'txtMail' => 'required|email',
103         'txtMdp' => 'required'
104     ]);
105
106     // Création et sauvegarde du client en BDD
107     $data = new Client();
108     $data->email = $request->txtMail;
109     $data->motpasse = $request->txtMdp; // Note : Idéalement, hacher le mot de passe ici
110     $data->nom = $request->txtNom;
111     $data->prenom = $request->txtPrenom;
112     $data->adresse = $request->txtAdresse;
113     $data->cp = $request->txtCp;
114     $data->ville = $request->txtVille;
115     $data->telephone = $request->txtTel;
116
117     $data->save();
118
119     // Connexion automatique après inscription (mise en session)
120     session(['client' => $data]);
121
122     return redirect()->route('accueil');
123 }
124
```

III. Planification du projet



La planification du projet est établie en plusieurs étapes (8 étapes différentes). Les étapes jaune-verte ont été faite par mon collègue (Malhon) et les étapes rouges ont été faite par moi (Kyllian).

IV. Conclusion

Le projet a été réalisé jusqu'au bout mais avec quelques difficultés au début sur la fonctionnalité 1 et 2. On s'est beaucoup entr aidé durant ce projet. Les tâches ont été répartis dès le début et sans ambiguïté. L'IA nous a aussi aidé pour régler des problèmes dont on ne trouver pas la solution. Le projet s'est très bien fini et dans de bonne circonstance.