

BONJOUR!

Je suis **Bruno RIGAL**

Etudiant en développement Web et Web mobile

Montpellier

DOSSIER PROFESSIONNEL

Développeur Web et Web Mobile

Bruno RIGAL

AutoRent - Location de véhicules en ligne

INTRODUCTION



Présentation du projet

AutoRent est une plateforme web de location de véhicules développée dans le cadre de ma formation de Développeur Web et Web Mobile.

Objectifs de cette présentation

- Démontrer mes compétences techniques
- Présenter l'architecture du projet
- Détails les fonctionnalités développées
- Expliquer les solutions aux défis rencontrés

Technologies utilisées

- Framework : Symfony (PHP)
- Base de données : MySQL
- Frontend : HTML, CSS, JavaScript, bootstrap
- Architecture : MVC

ARBORESCENCE DU PROJET - Vue d'ensemble

```
AutoRent/
  assets/
    styles/
      app.css
      scroll-animations.css # - Effets scroll snap scroll auto-freiné
      card-effects.css # - Animations cards hover
      transitions.css # - Transitions texte/éléments
    scripts/
      app.js
      scroll-snap.js # - Gestion scroll snap ANIMATIONS et INTERACTIONS
      intersection-observer.js # - Déetecte quand section est visible au scroll
      navbar-effects.js # - Navbar fixe dynamique
  bin/
    console
  config/
    packages/
      mailer.yaml # - Config email
      twig.yaml
      turbo.yaml # - Config evite l'écran blanc quand la page se charge
    routes/
      routes.yaml
      actualite_routes.yaml # Routes indique les routes à symfony
      admin_routes.yaml # - Routes actualités
      services.yaml
  migrations/
    Version20250125000001.php # Structure initiale
    Version20250126000002.php # Entités User/Véhicule
    Version20250127000003.php # Système réservation
    Version20250128000004.php # - Entité Actualité
```

SORTIE CONSOLE DE DÉBOGAGE TERMINAL PORTS

```
public/
  css/
    navbar.css
    responsive.css
  build/ # ASSETS COMPILÉS WEBPACK
  js/
    scroll-effects.css # - Effets scroll
    animations.js # - Animations
    vendors.js # Librairies externes
    runtime.js # Runtime Webpack
    actualites-css/
      scroll-effects.css # STYLES PAGE ACTUALITÉS
      - Animations scroll
    actualites-js/
      scroll-effects.js # SCRIPTS PAGE ACTUALITÉS
      - FONCTIONNEL - Observer animations
    css/
      home-css
        vehicles-list.css # STYLES PERSONNALISÉS
        # Styles modernes page véhicules
        navbar-fixed.css # - Navbar fixe responsive
        card-hover-effects.css # - Effets hover avancés
        transitions-global.css # - Transitions réutilisables
        home.js # - Gestion scroll accueil
    js/
      card-hover.js # - Effets hover cards
      text-animations.js # - Animations texte
      navbar-scroll.js # - Navbar réactive scroll
```

A

```
smooth-transitions.js # - Transitions pages
images/
  vehicules/
    thumbnails/
      full/
    actualites/
      featured/
      gallery/
    backgrounds/
    icons/
    logos/
    ui/
  uploads/
    actualites/
    vehicules/
  index.php
```

C

```
src/
  Controller/
    Admin/
      DashboardController.php # - Tableau de bord
      ReservationCrudController.php
      ActualiteAdminController.php # - Backend actualités
      VehiculeAdminController.php # - Gestion véhicules
      UserAdminController.php # - Gestion utilisateurs

      VehiculeController.php # - véhicules
      ActualiteController.php # - actualités
      ReservationController.php # - réservations
      ActualiteController.php # - Contrôleur front actualités
    CategorieController.php
    ContactController.php
    HomeController.php
    RegistrationController.php
    SecurityController.php
    VehiculeController.php
    SearchController.php
  DataFixtures/
    UserFixtures.php
    CategorieFixtures.php
    VehiculeFixtures.php
    ActualiteFixtures.php
    ReservationFixtures.php # - Méthode list() recherche
    # - Recherche globale
    # - DONNÉES DE TEST
  Entity/
    Actualite.php
    Categorie.php
    Reservation.php
    User.php
    Vehicule.php
    Media.php # - Données test actualités
    # - Réservations test
  Form/
    ActualiteFormType.php
    ContactFormType.php
    RegistrationFormType.php
    ReservationFormType.php
    VehiculeSearchType.php
    MediaUploadType.php # - Formulaire actualités
    # - Validation dates
    # - Formulaire recherche
    # - Upload médias
  Repository/
    ActualiteRepository.php # - Requêtes actualités
```

D

ARBORESCENCE DU PROJET - Backend & Templates

```
CategorieRepository.php  
ReservationRepository.php # - Méthodes disponibilité  
UserRepository.php  
VehicleRepository.php # rechercher des véhicules selon plusieurs critères  
MediaRepository.php # - Media Queries  
  
Security/  
    └─ CustomAuthenticator.php  
  
Service/  
    └─ NotificationService.php # - Service email  
        └─ ScrollAnimationService.php # - Gestion animations scroll  
    └─ ActualiteService.php # - Service actualités  
        └─ FileUploadService.php # - Upload fichiers  
        └─ ImageProcessingService.php # - Traitement images  
        └─ StatisticsService.php # - chiffres et graphiques pour l'admin  
        └─ SearchService.php # - Recherche avancée  
  
Eventlistener/  
    └─ ScrollEffectListener.php # - détecte les événements de scroll  
        └─ ExceptionListener.php # - Gestion erreurs  
        └─ SecurityListener.php # - Événements sécurité  
  
Command/  
    └─ GenerateReportCommand.php # - Génération rapports  
        └─ CleanupFilesCommand.php # - Nettoyage fichiers  
  
Twig/  
    └─ ScrollExtension.php # - Fonctions scroll  
        └─ DateExtension.php # - Formatage dates  
  
templates/  
    base.html.twig # Layout principal + navbar fixe  
    partials/  
        └─ navbar.html.twig # - Navbar fixe avec scroll  
            footer.html.twig  
            scroll-animations.html.twig # - Scripts animations  
            card-effects.html.twig # - Effets cards réutilisables  
            loading-spinner.html.twig # - Indicateur chargement  
            breadcrumbs.html.twig # - Fil d'Ariane chemin de navigation  
    emails/  
        layout.html.twig # Layout emails  
        admin_notification.html.twig # Email admin  
        user_confirmation.html.twig # Email client  
        reminder.html.twig # Email rappel  
    errors/  
        404.html.twig # Page 404 personnalisée client  
        500.html.twig # Page 500 personnalisée serveur  
    home/  
        index.html.twig # - Scroll snap + animations  
actualite/  
    index.html.twig # - ACTUALITÉS  
    show.html.twig # - Page actualités  
    category.html.twig # - Détail actualité  
    category.html.twig # - Actualités par catégorie  
admin/  
    layout.html.twig  
    dashboard.html.twig  
    actualite/  
        index.html.twig # - Liste admin actualités  
        edit.html.twig # - Édition actualité  
        new.html.twig # - Créditation actualité  
        show.html.twig # - Détail admin  
        vehicle/  
            index.html.twig  
            edit.html.twig  
            new.html.twig # - Gestion véhicules
```

E

```
reservation/  
    └─ index.html.twig # - Gestion réservations  
        edit.html.twig  
categorie/  
    └─ show.html.twig # - Cards avec effets shadow  
vehicle/  
    └─ index.html.twig # Liste véhicules  
        show.html.twig # - Date disponibilité  
        list.html.twig # - Recherche + résultats  
reservation/  
    └─ index.html.twig # Formulaire réservation  
        confirmation.html.twig # - Page confirmation  
        history.html.twig # - Historique utilisateur  
components/  
    └─ card-vehicle.html.twig # Card véhicule avec effets  
        scroll-section.html.twig # - SECTION avec effet scroll snap (accrochage automatique)  
        text-animation.html.twig # - Texte avec animation  
        pagination.html.twig # - Pagination personnalisée  
        search-filters.html.twig # - Filtres de recherche  
security/  
    └─ login.html.twig  
        forgot_password.html.twig # - Mot de passe oublié  
registration/  
    └─ register.html.twig  
profile/  
    └─ show.html.twig # Profil utilisateur  
        edit.html.twig # Édition profil  
contact/  
    └─ index.html.twig  
  
tests/  
    Functional/  
        VehiculeControllerTest.php # TESTS FONCTIONNELS  
        ReservationFlowTest.php # Tests end-to-end  
        ActualiteControllerTest.php # Test complet réservation  
        ScrollEffectsTest.php # - Tests animations  
        AdminWorkflowTest.php # - Tests administration  
        UserJourneyTest.php # TESTS UNITAIRES  
    Unit/  
        Controller/  
            CategorieControllerTest.php  
            ContactControllerTest.php  
            HomeControllerTest.php # - Tests scroll snap  
            RegistrationControllerTest.php  
            ReservationControllerTest.php  
            SecurityControllerTest.php  
            VehiculeControllerTest.php  
        Service/  
            NotificationServiceTest.php # Test service email  
            ActualiteServiceTest.php # - Tests service actualités  
            ScrollAnimationTest.php # - Tests animations  
            FileuploadServiceTest.php # - Tests upload  
            StatisticsServiceTest.php # - Tests statistiques  
        Entity/  
            ActualiteTest.php # Tests entité actualité  
            UserTest.php  
            VehiculeTest.php  
            ReservationTest.php  
        Repository/  
            ActualiteRepositoryTest.php # Tests requêtes  
            VehiculeRepositoryTest.php  
    Integration/ # - TESTS INTÉGRATION
```

F

G

H

ARBORESCENCE DU PROJET - Configuration & Bilan

```
|- DatabaseTest.php          # Tests base de données  
|- fixtures/  
|   |- images/  
|       |- data/  
|- translations/  
|   |- messages.fr.yaml     # - Traductions françaises  
|   |- validators.fr.yaml   # - Messages validation  
|   |- admin.fr.yaml        # - Interface admin  
|   |- emails.fr.yaml       # - Emails français  
|- vendor/  
|- docs/  
  
|- .env                      # VARIABLES ENVIRONNEMENT  
|- .env.local                 # ENVIRONNEMENT TEST  
|- .gitignore                # Git ignore rules  
|- webpack.config.js         # BUILD ASSETS  
|- phpunit.xml.dist          # CONFIGURATION TESTS  
|- phpcs.xml                 # - Code style PHP  
|- Makefile                  # - Commandes make  
|- README.md                 # DOCUMENTATION PROJET
```

STATISTIQUES FINALES DU PROJET :

- 8 NOUVEAUX STYLES CSS
- 7 NOUVEAUX Scripts JS
- Système actualités complet avec animations
- Effets scroll snap implémentés
- Animations cards/texte avancées
- Navbar fixe responsive
- Interface admin complète
- Système notifications email
- Interface multilingue
- Design responsive complet

SOMMAIRE

- PARTIE 1

1. Développer la partie front-end d'une application web ou web mobile sécurisée

- a. Installer et configurer son environnement de travail en fonction du projet web ou web mobile
- b. Maquetter des interfaces utilisateur web ou web mobile
- c. Réaliser des interfaces utilisateur web ou web mobile
- d. Développer la partie dynamique des interfaces utilisateur web ou web mobile

SOMMAIRE

- PARTIE 2

1. Développer la partie back-end d'une application web ou web mobile sécurisée

e. Mettre en place une base de donnée relationnelle

f. Développer des composants d'accès aux données SQL et NoSQL

g. Développer des composants métier côté serveur

h. Documenter le déploiement d'une application dynamique web ou web mobile

CONTEXT TECHNIQUE ET METHODOLOGIE

Environnement de développement

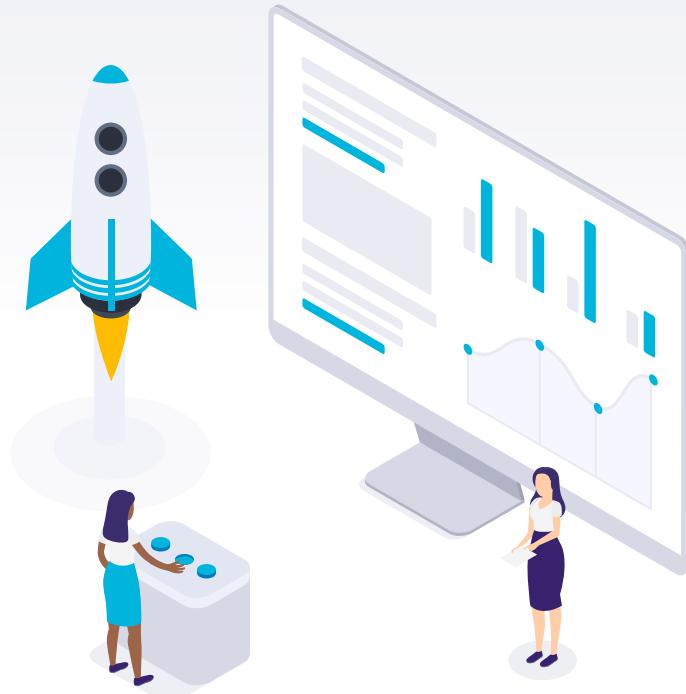
- VS Code - Éditeur de code principal
- Extensions : PHP Intelephense, Symfony
- Auto-complétion et coloration syntaxique

Contexte du projet

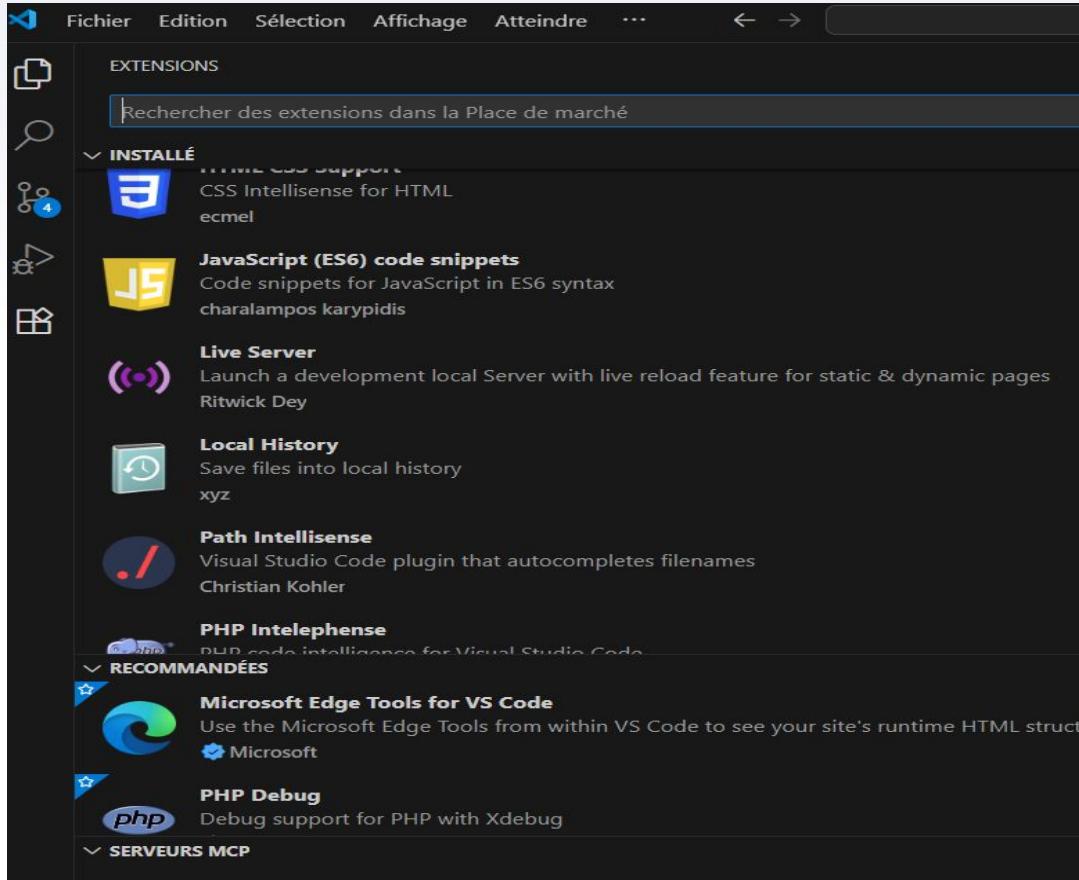
- Projet réalisé en solo
- Pas de communication d'équipe
- Documentation personnel

Méthodologie de travail

- Organisation: Tableau Kanban
- Test réguliers



ENVIRONNEMENT VS CODE



Extensions installées pour le développement

COMMANDES GIT - VERSIONING

Commandes de base

- git init - Initialiser un repository Git
- git add . - Ajouter tous les fichiers modifiés
- git commit -m "message" - Enregistrer
- git push origin main - Envoyer le code vers GitHub
- git clone - récupérer un projet qui a été sauvegardé dans github

Gestion des branches

- git checkout -b feature - Créer branche
- git merge feature - Fusionner

Synchronisation

- git pull origin main - Récupérer
- git status - Vérifier état

HISTORIQUE DES COMMITS - GITHUB

Commits on Sep 26, 2025
<div><p>mise en place d'images multiple dans détail avec son carrousel approprié</p><p>rigal34 committed 3 weeks ago</p><p>Unverified f42eb68</p></div>
Commits on Sep 14, 2025
<div><p>mise en place d'images multiple dans détail avec son carrousel approprié</p><p>rigal34 committed on Sep 14</p><p>Unverified Sad1606</p></div>
Commits on Sep 13, 2025
<div><p>ajout des images principales dans les cartes de mon catalogue</p><p>rigal34 committed on Sep 13</p><p>Unverified 08338f5</p></div>
Commits on Sep 11, 2025
<div><p>ajout d'un fichier responsive pour gérer les media query</p><p>rigal34 committed on Sep 11</p><p>creation d'un fichier CSS pour le menu</p><p>rigal34 committed on Sep 11</p><p>Unverified 671080e</p></div>
<div><p>mise en place de dossier et fichier home.js et home.css</p><p>rigal34 committed on Sep 11</p><p>Unverified fb16908</p></div>
<div><p>feat(actualités): système d'animations scroll complet</p><p>rigal34 committed on Sep 11</p><p>Unverified d76465f</p></div>
Commits on Sep 8, 2025
<div><p>feat(ui): scroll snap navigation + shadow effects</p><p>rigal34 committed on Sep 8</p><p>Unverified bca4abb</p></div>
Commits on Sep 7, 2025
<div><p>ajout de champs commentaires et d'un champ de demande de permis pour formulaire de réservation</p><p>rigal34 committed on Sep 7</p><p>Unverified 599187e</p></div>
Commits on Sep 6, 2025
<div><p>Add: Système de notification par email pour les réservations</p><p>rigal34 committed on Sep 6</p><p>Unverified fc0f55a</p></div>
Commits on Sep 5, 2025
<div><p>feat: Tests unitaires/fonctionnels + Interface moderne pour les véhicules</p><p>rigal34 committed on Sep 5</p><p>Unverified 5046119</p></div>
Commits on Aug 30, 2025
<div><p>ajout d'un menu de recherche dans mon catalogue</p><p>rigal34 committed on Aug 30</p><p>Unverified 510e1b6</p></div>
Commits on Aug 29, 2025
<div><p>mise en place de la disparition du bouton de déconnexion</p><p>rigal34 committed on Aug 29</p><p>Unverified 19ab0ef</p></div>
Commits on Aug 27, 2025

Travail régulier et messages descriptifs
Utilisation de Git et GitHub tout au long du projet
pour versionner le code, suivre l'évolution et
collaborer efficacement.

Gestion de versions avec Git

- Commits réguliers et fréquents
- Messages descriptifs et clairs
- Historique complet du développement
- Traçabilité des modifications

GitHub - Plateforme collaborative

- Repository distant pour sauvegarde
- Branches pour features (frontend, backend)
- Commits organisés par fonctionnalité
- Gestion de projet professionnelle

STACK TECHNIQUE - VERSIONS

Backend

- PHP 8.2
- Symfony 6.4
- MySQL 8.0
- Doctrine ORM

Frontend

- HTML5 / CSS3
- JavaScript ES6
- Bootstrap 5
- CSS Media Queries
- Twig 3.x

Outils de développement

- Git 2.42
- VS Code
- Symfony CLI
- Composer 2.6
- npm 9.8
- XAMPP 8.2
- SQLite

VERSIONS INSTALLÉES

```
PHP 8.2.12 (cli) (built: Oct 24 2023 21:15:15) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.12, Copyright (c) Zend Technologies
PS C:\xampp\htdocs\AutoRent> composer -V
PHP 8.2.12 (cli) (built: Oct 24 2023 21:15:15) (ZTS Visual C++ 2019 x64) ...
```

- PS C:\xampp\htdocs\AutoRent> composer -V
Composer version 2.8.6 2025-02-25 13:03:50
PHP version 8.2.12 (C:\xampp\php\php.exe)
Run the "diagnose" command to get more detailed diagnostics output.
- PS C:\xampp\htdocs\AutoRent> symfony version

INFO A new Symfony CLI version is available (5.15.1, currently running 5.11.0).

If you installed the Symfony CLI via a package manager, updates are going to be automatic.
If not, upgrade by downloading the new version at <https://github.com/symfony-cli/symfony-cli/releases>
And replace the current binary (`symfony.exe`) by the new one.

- Symfony CLI version 5.11.0 (c) 2021-2025 Fabien Potencier (2025-02-17T09:11:09Z - stable)
- PS C:\xampp\htdocs\AutoRent> node -v
v22.18.0
- PS C:\xampp\htdocs\AutoRent> npm -v
10.9.3
- PS C:\xampp\htdocs\AutoRent> git --version
git version 2.48.1.windows.1
- PS C:\xampp\htdocs\AutoRent> █

Terminal VS Code - Versions de l'environnement

CAHIER DES CHARGES FONCTIONNEL

Démarche adoptée

- Identification des 3 types d'utilisateurs
- Visiteur (non connecté)
- Utilisateur connecté (client)
- Administrateur (gestion du parc)

- Méthode : User Stories Exemple :
"En tant que Visiteur, je veux consulter le catalogue pour choisir un véhicule"

ID	(Rôle)	Fonction Fonctionnalités	Raisons/Objectifs	Critères d'Acceptation
1	En tant que VISITEUR	Je veux Voir le catalogue	Afin de (pour) Voir une liste paginée des véhicules avec leur photo, nom et tarif	La page affiche 4 véhicules maximum. On peut naviguer entre les pages. Chaque véhicule est cliquable.

FONCTIONNALITÉS PRINCIPALES

Catalogue de véhicules

- Affichage en grille avec photo, nom et prix/jour
- Bouton "Voir détails" pour chaque véhicule
- Page détail avec informations complètes du véhicule

Actualités

- Page dédiée aux actualités de l'entreprise
- Gestion et publication par l'administrateur
- Affichage chronologique des articles

Contact

- Formulaire de contact

SYSTÈME DE RÉSERVATION

Processus de réservation

- Vérification disponibilité
- Authentication utilisateur
- Formulaire complet

Calcul et validation

- Calcul auto du prix
- Véhicule indisponible
- Notification email

Gestion des réservations

- Statut "en attente"
- Gestion administrative

TABLEAU USER STORIES (1/2)

1	En tant que	VISITEUR	Je veux	Voir le catalogue	Afin de (pour)	Voir une liste paginée des véhicules avec leur photo, nom et tarif	La page affiche 4 véhicules maximum. On peut naviguer entre les pages. Chaque véhicule est cliquable.
2	En tant que	VISITEUR	Je veux	Voir le détail d'un véhicule	Afin de (pour)	Cliquer sur un véhicule pour voir sa page détaillée	La page affiche toutes les infos du cahier des charges. La galerie d'images fonctionne.
3	En tant que	VISITEUR	Je veux	M'inscrire avec un email et un mot de passe sécurisé	Afin de (pour)	Créer un compte pour pouvoir réserver	Le mot de passe doit être validé (ex: 8 caractères min). L'email doit être unique. Un message de succès s'affiche.
4	En tant que	UTILISATEUR	Je veux	Me connecter	Afin de (pour)	Accéder à mon espace personnel et commencer une réservation	Si les identifiants sont bons, je suis redirigé vers mon tableau de bord. Sinon, un message d'erreur apparaît.
5	En tant que	UTILISATEUR Déconnecté	Je veux	Me déconnecter	Afin de (pour)	Vider ma session	Je suis immédiatement redirigé vers la page d'accueil, avec un message de déconnexion
6	En tant que	UTILISATEUR Connecté	Je veux	Louer un véhicule	Afin de (pour)	Préparer ma réservation avant de la confirmer	Le panier se met à jour avec le véhicule, les dates et le prix. Le coût total est recalculé.
7	En tant que	UTILISATEUR Connecté	Je veux	Accéder à une page récapitulant mon panier	Afin de (pour)	Vérifier ma commande avant de payer	La page affiche toutes les lignes de réservation. Je peux modifier ou supprimer une ligne. Le total est clairement visible.
8	En tant que	UTILISATEUR Connecté	Je veux	Consulter ses réservations dans son espace client	Afin de (pour)	Finaliser ma location	Le formulaire de conducteur est validé. La disponibilité est re-vérifiée. Un email de confirmation est envoyé.
9	En tant que	UTILISATEUR connecté	Je veux	Consulter l'historique de mes réservations dans mon espace client	Afin de (pour)	Suivre mes locations passées et à venir	Je peux cliquer pour voir les détails.
10	En tant que	VISITEUR	Je veux	Utiliser une barre de recherche pour trouver un véhicule par son nom	Afin de (pour)	Trouver rapidement un modèle spécifique	La recherche se fait sur le nom du véhicule. La liste des résultats est mise à jour.
11	En tant que	ADMINISTRATEUR	Je veux	Remplir un formulaire pour ajouter un nouveau véhicule au parc	Afin de (pour)	L'intégrer au catalogue et le rendre disponible à la location	Quand je clique sur le bouton Ajouter un nouveau véhicule

Partie 1/2 - Organisation des besoins

TABLEAU USER STORIES (2/2)

11	En tant que	ADMINISTREUR	Je veux	Remplir un formulaire pour ajouter un nouveau véhicule au parc	Afin de (pour)	L'intégrer au catalogue et le rendre disponible à la location	Quand je clique sur le bouton Ajouter un nouveau véhicule	Haute	8
12	En tant que	ADMINISTREUR	Je veux	Pouvoir ajouter une ou plusieurs nouvelles images à la galerie d'un véhicule	Afin de (pour)	Enrichir la fiche produit et aider les clients à mieux visualiser le véhicule	Quand je clique sur le champ Fichier et que je sélectionne une ou plusieurs images	Moyen	13
13	En tant que	ADMINISTREUR	Je veux	Supprimer une ou plusieurs images de la galerie d'un véhicule	Afin de (pour)	Garder la présentation des véhicules propre et à jour	Quand je clique sur son icône de suppression	Moyen	5
14	En tant que	ADMINISTREUR	Je veux	Accéder à une page listant toutes les réservations effectuées sur le site	Afin de (pour)	Avoir une vue d'ensemble de l'activité et pouvoir gérer les contrats.	Quand je clique sur un bouton pour afficher les réservations	Haute	2
15	En tant que	ADMINISTREUR	Je veux	Pouvoir approuver une réservation qui est en attente de validation"	Afin de (pour)	Confirmer officiellement la location au client et mettre à jour le planning des véhicules.	Quand je clique sur un bouton Approuver	Haute	3
16	En tant que	ADMINISTREUR	Je veux	Pouvoir modifier les rôles d'un utilisateur existant depuis le back-office	Afin de (pour)	Lui donner des droits d'administration ou les lui retirer de manière simple et sécurisée	je vois une section Rôles avec une ou plusieurs cases à cocher.	Haute	5
17	En tant que	VISITEUR	Je veux	utiliser un système de filtre par catégories	Afin de (pour)	pouvoir filtrer les véhicules par une ou plusieurs catégories	Faciliter ma recherche en affinant les résultats par types de véhicules qui m'intéressent, même s'ils ont plusieurs classification	Haute	3
18	En tant que	ADMINISTREUR	Je veux	pouvoir clôturer une réservation terminé	Afin de (pour)	Finaliser le contrat de location et rendre le véhicule de nouveau disponible dans le parc.	Alors les réservations dont la date de fin est passée ont un statut clair	Haute	3
19	En tant que	ADMINISTREUR	Je veux	Je veux pouvoir associer un véhicule à plusieurs catégories existantes	Afin de (pour)	Classer précisément chaque véhicule selon toutes ses caractéristiques (ex: "Citadine" ET "Électrique") et faciliter la recherche des utilisateurs.	Sur la page d'édition d'un véhicule dans le back-office, je vois un champ qui me permet de sélectionner plusieurs catégories pour ce véhicule (ex: liste à choix multiples, ou tags). Les modifications sont bien enregistrées en base de données.	Haute	13
20	En tant que	ADMINISTREUR	Je veux	Je veux pouvoir marquer un véhicule comme "disponible" ou "indisponible"	Afin de (pour)	Contrôler la visibilité d'un véhicule dans le catalogue sans le supprimer, par exemple s'il est en maintenance ou loué.	Sur la page d'édition ou de liste d'un véhicule dans le back-office, je vois une option simple (ex: case à cocher "Disponible" ou interrupteur) pour modifier son statut. Les véhicules marqués "indisponible" ne sont pas affichés dans le catalogue public.	Haute	5
21	En tant que	ADMINISTREUR	Je veux	Je veux Gérer (créer, modifier, supprimer) des articles d'actualité	Afin de (pour)	Tenir les visiteurs et utilisateurs informés des nouveautés, des promotions ou des événements liés à AutoRent.	Je peux accéder à une section "Actualités" dans le back-office.	Moyen	8

Partie 2/2 - Gestion et administration

MAQUETTAGE AVEC FIGMA

Outil de conception

- Figma
- Design collaboratif et responsive
- Maquettage de toutes les interfaces du site

Approche Mobile-first

- Mobile : 431px (iphone)
- Tablette : 768px (ipad)
- Desktop : 1440 (PC)

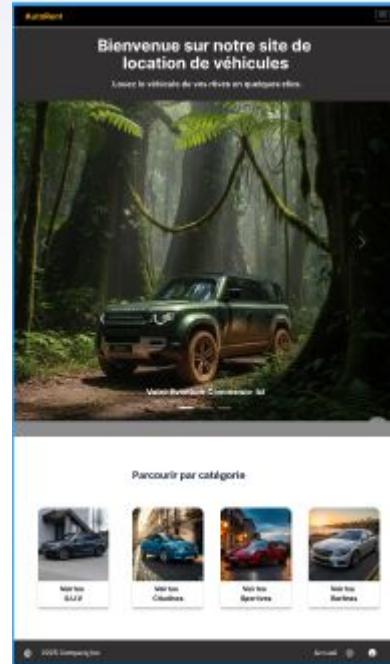
Outils Figma utilisés

- Wireframe (structure)
- Frame (cadres)
- Component (composants)
- Groupes

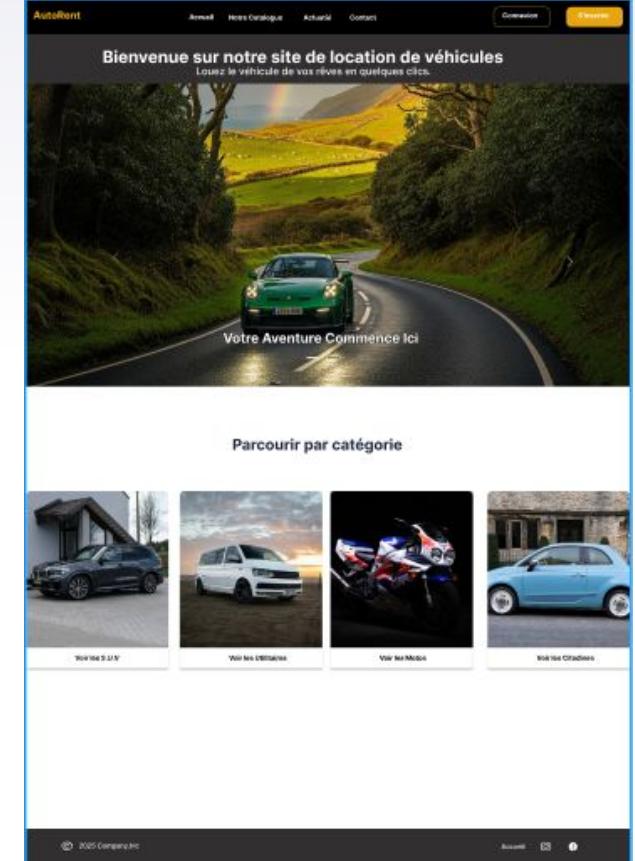
APERÇU MAQUETTES FIGMA



Tablet - Accueil </>



Desktop - Accueil </>



Page statique et responsive

CODE HTML BOOTSTRAP

```
5 <nav class="navbar navbar-expand-lg navbar-dark fixed-top navbar-prestige">
6   <div class="container-fluid">
7     <a class="navbar-brand" href="{{ path('app_home') }}> AutoRent</a>
8
9     <button class="navbar-toggler" type="button"
10        data-bs-toggle="collapse"
11        data-bs-target="#navbarNav"
12        aria-controls="navbarNav"
13        aria-expanded="false"
14        aria-label="Toggle navigation">
15       <span class="navbar-toggler-icon"></span>
16     </button>
17
18     <div class="collapse navbar-collapse" id="navbarNav">
19       <ul class="navbar-nav mx-auto">
20         <li class="nav-item">
21           <a class="nav-link text-white" href="{{ path('app_home') }}>Accueil</a>
22         </li>
```

Interface statique HTML/CSS

- HTML5 - Balises sémantiques (<nav>, ,)
- Code statique avec template Twig
- Bootstrap 5 via Webpack Encore (npm)

Bootstrap responsive

- navbar navbar-expand-lg → Navigation responsive
- navbar-toggler → Bouton menu burger (mobile)
- collapse navbar-collapse → Menu masquable
- container-fluid → Container adaptatif pleine largeur
- Desktop (>992px) : Menu horizontal complet
- Mobile (<992px) : Menu burger cliquable
- Transitions automatiques Bootstrap

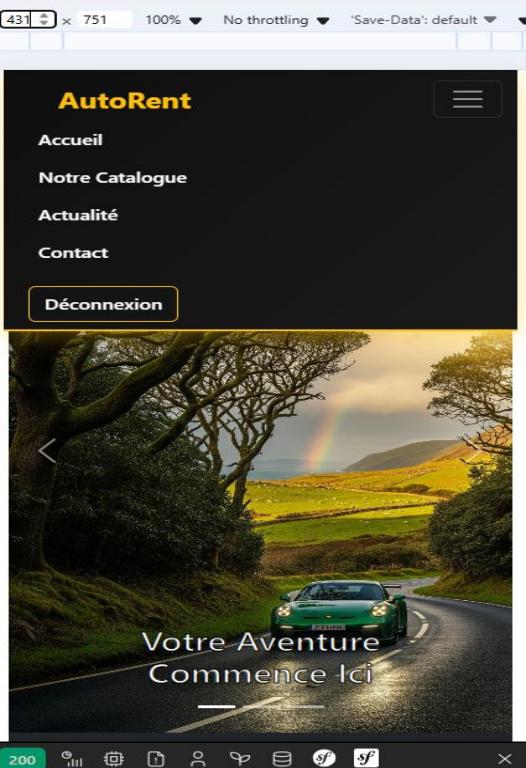
SEO Optimisation pour les moteurs de recherche

Partie référencement et ses balises sémantiques

```
templates > base.html.twig
1  <!DOCTYPE html>
2  <html lang="fr">
3
4      <head>
5          <meta charset="UTF-8">
6          <meta name="viewport" content="width=device-width, initial-scale=1.0">
7          <meta name="description" content="Site de location de voitures.">
8          <meta name="description" content=""
9          {% block meta_description %}
10             AutoRent : Location de voitures à prix compétitifs.
11             Large choix de véhicules économiques, berlines, SUV et voitures de luxe.
12             Réservation en ligne simple et rapide. Devis gratuit.
13             {% endblock %}
14         ">
15
16         <title>
17             {% block title %}AutoRent - Location de voitures à prix compétitifs
18             {% endblock %}
19         </title>
```

RENDU RESPONSIVE - 2 FORMATS SUR NAVIGATEUR

Page d'accueil AutoRent



A screenshot of a desktop browser window showing the same homepage. The header "AutoRent" is at the top left, followed by a navigation bar with links: Accueil, Notre Catalogue, Actualité, Contact, and Déconnexion. The main content area features a large photograph of a green sports car driving on a winding road through a lush green landscape. Overlaid on the center of the photo is the text "sur notre site de location de véhicules" and "Louez le véhicule de vos rêves en quelques clics.". Below the photo is another overlay with the text "Votre Aventure Commence Ici". The browser's address bar shows dimensions of 1440 x 651, and the status bar at the bottom includes icons for zoom, orientation, and battery.

Interface adaptative - HTML5, CSS3, Bootstrap 5

CODE CSS - MOBILE FIRST

Fichier : responsive.css AutoRent

```
10  /* CAROUSEL DE LA PAGE D'ACCUEIL - Mobile par défaut */
11 .carousel-inner img {
12   height: 250px;
13 }
14
15 /* CAROUSEL DE LA PAGE DE DÉTAIL - Mobile par défaut */
16 .carousel-image {
17   height: 200px;
18 }
19
20 /* HERO SECTION - Mobile par défaut */
21 .hero-section header {
22   min-height: 60vh; /* Valeur confortable pour mobile */
23 }
24
25
26 /* TABLETTE (à partir de 769px)
27 @media (min-width: 769px) {
28
29   .carousel-inner img {
30     height: 350px;
31   }
32
33   .carousel-image {
34     height: 300px;
35   }
36
37   .hero-section header {
38     min-height: 55vh;
39   }
40 }
```

ÉCRANS LARGES MAIS PAS TRÈS HAUTS

```
@media (min-width: 1200px) and (max-height: 800px) {
  .hero-section header {
    min-height: 50vh;
  }

  .carousel-inner img {
    height: 480px;
  }
}
```

ÉCRANS TRÈS LARGES ET PAS TRÈS HAUTS

```
@media (min-width: 1400px) and (max-height: 700px) {
  .hero-section header {
    min-height: 45vh;
  }

  .carousel-inner img {
    height: 420px;
  }
}
```

J'ai commencé par du mobile first une approche progressive et ensuite par les media query en version tablette, pour terminer par les écrans plus grands

Les raisons de ces choix

- SEO
- Temps de chargement réduit sur mobile
- Une grande partie du trafic web provient du mobile

GESTION DES ANIMATIONS D'APPARITION AU SCROLL avec javascript

```
v function createObserver() {  
v   if (observer) {  
v     observer.disconnect();  
v   }  
  
v   observer = new IntersectionObserver((entries) => {  
v     entries.forEach(entry => {  
v       if (entry.isIntersecting) {  
v         entry.target.classList.add('show');  
v       } else {  
v         entry.target.classList.remove('show');  
v       }  
v     });  
v   }, {  
v     threshold: 0.1, // sensible déclencheur apparition pour le texte en haut  
v     rootMargin: '50px 0px -30px 0px' // Déclenche l animation pour évite des effets visuels saccadés  
v   });  
v }
```

- Je crée un observateur
- Il surveille quand un élément est visible
- Il ajoute/retire une classe selon la visibilité
- Création de l'objet
- paramètre automatiquement injecté par l'API
- parcourt/boucle sur chaque élément
- Quand l'élément est visible on ajoute la classe
- Quand l'élément n'est pas visible on retire la classe

TEMPLATES DYNAMIQUES - TWIG

home/index.html.twig

```
<section class="categories section d-flex flex-column justify-content-center">
    <div class="container-fluid">
        <!-- TITRE AVEC CLASSE POUR ANIMATION -->
        <div class="row">
            <div class="col-12">
                <h2 class="text-center mb-5 fw-bold category-title">Parcourir par catégorie</h2>
            </div>
        </div>

        <!-- CARDS AVEC CLASSE POUR ANIMATION -->
        <div class="row justify-content-center align-items-center" id="cards-container">
            {% for categorie in categories %}
                <div class="col-md-3 mb-4 category-card">
                    <div class="card shadow-sm h-100">
                        
                        <div class="card-body text-center">
                            <a class="nav-link text-dark" href="{{ path('app_categorie_show', {'id': categorie.id}) }}><strong>Voir les {{ categorie.nom }}</strong></a>
                        </div>
                    </div>
                </div>
            {% endfor %}
        </div>
    </div>
</section>
```

Templates dynamiques

- Moteur de templates Symfony
- Boucle `{% for %}` pour listes
- Variables `{{ variable }}` dynamiques

Génération contenu dynamique

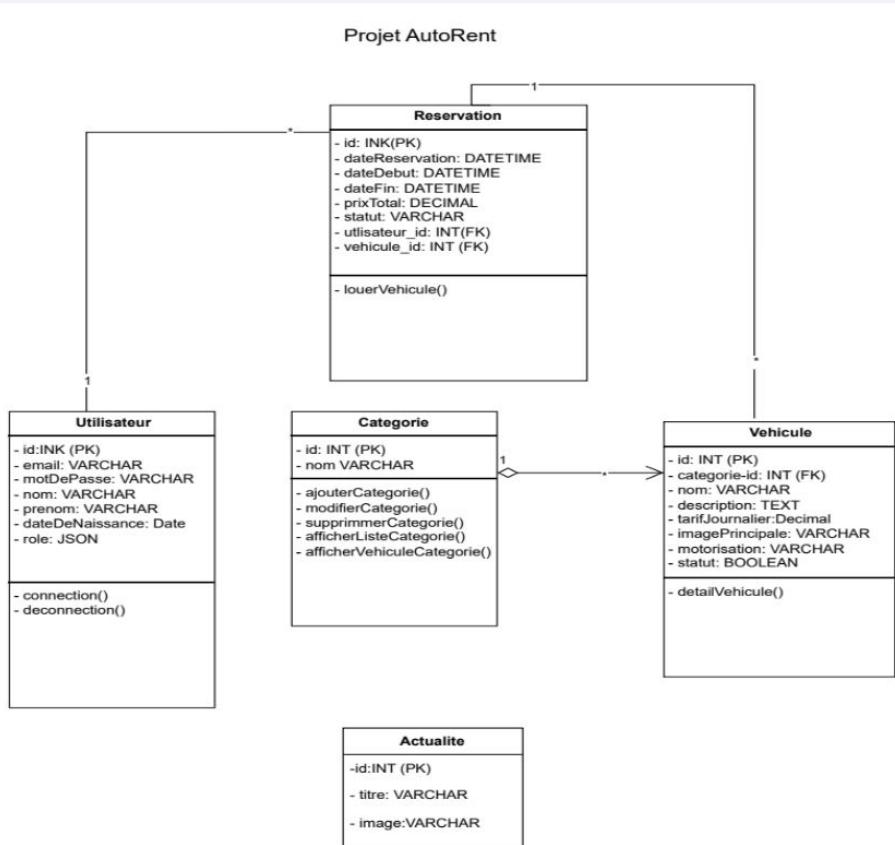
- Le contenu change automatiquement selon les données de la base de données

Avantages Twig

- récupère les données envoyé par le contrôleur
- Protection contre les attaques XSS
- Intégration Bootstrap + HTML5

MODÉLISATION UML - CONCEPTION BASE DE DONNÉES

Projet AutoRent - Diagramme de classes



Démarche UML

J'ai commencé mon projet par la modélisation UML pour avoir une vision claire et structurée de mon application avant de coder. L'UML m'a permis de réfléchir à l'architecture et aux relations entre mes différentes entités.

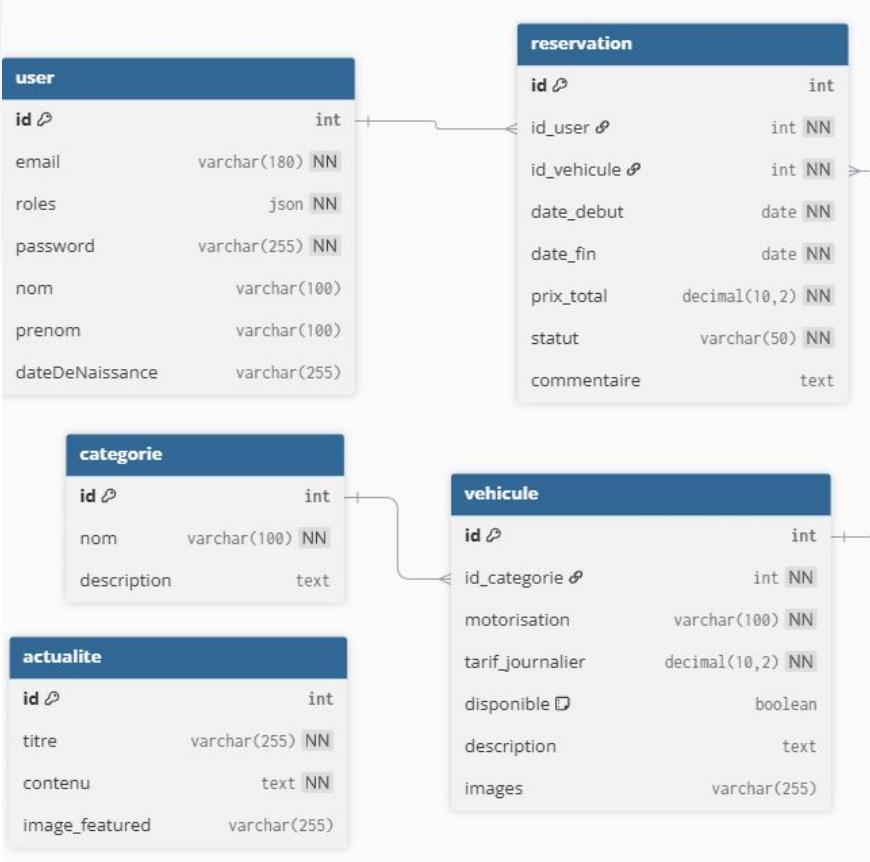
Relations

OneToMany : Categorie → Vehicule

ManyToOne : Reservation → User, Vehicule
Clés étrangères et cardinalités définies

MODÈLE PHYSIQUE DE DONNÉES (MPD)

Structure de la base de données AutoRent



Après avoir défini mes entités et leurs relations dans le diagramme UML, j'ai créé ce MPD pour traduire cette conception en instructions précises, pour la mise en préparation de la gestion de ma base de données. Mon MPD pour ce projet AutoRent comprend 5 tables principales

5 tables :

user, reservation, categorie, vehicule
actualite

STRUCTURE BASE DE DONNÉES - MySQL

Structure relationnelle

actualite		CREATE TABLE actualite (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, titre VARCHAR(255) NOT NULL, contenu CLOB NOT NULL, image VARCHAR(255) "image" VARCHAR(255) DEFAULT NULL)
	id	INTEGER "id" INTEGER NOT NULL
	titre	VARCHAR(255) "titre" VARCHAR(255) NOT NULL
	contenu	CLOB "contenu" CLOB NOT NULL
	image	VARCHAR(255) "image" VARCHAR(255) DEFAULT NULL
categorie		CREATE TABLE categorie (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, nom VARCHAR(255) NOT NULL, image VARCHAR(255) "image" VARCHAR(255) DEFAULT NULL)
	id	INTEGER "id" INTEGER NOT NULL
	nom	VARCHAR(255) "nom" VARCHAR(255) NOT NULL
	image	VARCHAR(255) "image" VARCHAR(255) DEFAULT NULL
messenger_messages		CREATE TABLE messenger_messages (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, body CLOB NOT NULL, created_at DATETIME NOT NULL, updated_at DATETIME NOT NULL, type VARCHAR(255) NOT NULL)
reservation		CREATE TABLE reservation (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, utilisateur_id INTEGER DEFAULT NULL, vehicule_id INTEGER NOT NULL, date_reservation DATETIME NOT NULL, date_debut DATETIME NOT NULL, date_fin DATETIME NOT NULL, prix_total NUMERIC(10, 2) "prix_total" NUMERIC(10, 2) NOT NULL, statut VARCHAR(255) "statut" VARCHAR(255) NOT NULL, type_permis VARCHAR(10) "type_permis" VARCHAR(10) DEFAULT NULL, commentaires CLOB DEFAULT NULL)
	id	INTEGER "id" INTEGER NOT NULL
	utilisateur_id	INTEGER "utilisateur_id" INTEGER DEFAULT NULL
	vehicule_id	INTEGER "vehicule_id" INTEGER NOT NULL
	date_reservation	DATETIME "date_reservation" DATETIME NOT NULL
	date_debut	DATETIME "date_debut" DATETIME NOT NULL
	date_fin	DATETIME "date_fin" DATETIME NOT NULL
	prix_total	NUMERIC(10, 2) "prix_total" NUMERIC(10, 2) NOT NULL
	statut	VARCHAR(255) "statut" VARCHAR(255) NOT NULL
	type_permis	VARCHAR(10) "type_permis" VARCHAR(10) DEFAULT NULL
	commentaires	CLOB "commentaires" CLOB DEFAULT NULL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)

user		CREATE TABLE user (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, email VARCHAR(180) NOT NULL, roles CLOB NOT NULL, password VARCHAR(255) "password" VARCHAR(255) NOT NULL, nom VARCHAR(255) "nom" VARCHAR(255) NOT NULL, prenom VARCHAR(255) "prenom" VARCHAR(255) NOT NULL, date_de_naissance DATE DEFAULT NULL)
	id	INTEGER "id" INTEGER NOT NULL
	email	VARCHAR(180) "email" VARCHAR(180) NOT NULL
	roles	CLOB "roles" CLOB NOT NULL
	password	VARCHAR(255) "password" VARCHAR(255) NOT NULL
	nom	VARCHAR(255) "nom" VARCHAR(255) NOT NULL
	prenom	VARCHAR(255) "prenom" VARCHAR(255) NOT NULL
	date_de_naissance	DATE "date_de_naissance" DATE DEFAULT NULL
vehicule		CREATE TABLE vehicule (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, nom VARCHAR(255) NOT NULL, description CLOB DEFAULT NULL, tarif_journalier NUMERIC(10, 2) "tarif_journalier" NUMERIC(10, 2) NOT NULL, image_principale VARCHAR(255) "image_principale" VARCHAR(255) NOT NULL, motorisation VARCHAR(255) "motorisation" VARCHAR(255) NOT NULL, images CLOB DEFAULT NULL, statut BOOLEAN "statut" BOOLEAN DEFAULT NULL)
	id	INTEGER "id" INTEGER NOT NULL
	nom	VARCHAR(255) "nom" VARCHAR(255) NOT NULL
	description	CLOB "description" CLOB DEFAULT NULL
	tarif_journalier	NUMERIC(10, 2) "tarif_journalier" NUMERIC(10, 2) NOT NULL
	image_principale	VARCHAR(255) "image_principale" VARCHAR(255) NOT NULL
	motorisation	VARCHAR(255) "motorisation" VARCHAR(255) NOT NULL
	images	CLOB "images" CLOB DEFAULT NULL
	statut	BOOLEAN "statut" BOOLEAN DEFAULT NULL
vehicule_categorie		CREATE TABLE vehicule_categorie (vehicule_id INTEGER NOT NULL, categorie_id INTEGER NOT NULL, PRIMARY KEY (vehicule_id, categorie_id))
	vehicule_id	INTEGER "vehicule_id" INTEGER NOT NULL
	categorie_id	INTEGER "categorie_id" INTEGER NOT NULL

RELATIONS BASE DE DONNÉES RELATIONNELLE

CREATE TABLE reservation (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,		
id	INTEGER	"id" INTEGER NOT NULL
utilisateur_id	INTEGER	"utilisateur_id" INTEGER DEFAULT NULL
vehicule_id	INTEGER	"vehicule_id" INTEGER NOT NULL
date_reservation	DATETIME	"date_reservation" DATETIME NOT NULL
date_debut	DATETIME	"date_debut" DATETIME NOT NULL
date_fin	DATETIME	"date_fin" DATETIME NOT NULL
prix_total	NUMERIC(10, ...)	"prix_total" NUMERIC(10, 2) NOT NULL
statut	VARCHAR(255)	"statut" VARCHAR(255) NOT NULL
type_permis	VARCHAR(10)	"type_permis" VARCHAR(10) DEFAULT NULL
commentaires	CLOB	"commentaires" CLOB DEFAULT NULL

CREATE TABLE vehicule_categorie (
vehicule_id	INTEGER	"vehicule_id" INTEGER NOT NULL
categorie_id	INTEGER	"categorie_id" INTEGER NOT NULL

Table RESERVATION:

- **utilisateur_id** → FK vers user
 - **vehicule_id** → FK vers vehicule
- Relations OneToMany** (1→*)

TABLE PIVOT vehicule_categorie :

- **vehicule_id** → FK vers vehicule
 - **categorie_id** → FK vers categorie
- Contient UNIQUEMENT les 2 FK
Permet **ManyToMany** (N:N)

COMPOSANTS D'ACCÈS AUX DONNÉES - Configuration

Fichier : config/packages/doctrine.yaml

```
doctrine:  
    dbal:  
        url: '%env(resolve:DATABASE_URL)%'  
  
        # IMPORTANT: You MUST configure your server version,  
        # either here or in the DATABASE_URL env var (see .env file)  
        #server_version: '16'  
  
        profiling_collect_backtrace: '%kernel.debug%'  
        use_savepoints: true  
  
    orm:  
        auto_generate_proxy_classes: true  
        enable_lazy_ghost_objects: true  
        report_fields_where_declared: true  
        validate_xml_mapping: true  
        naming_strategy: doctrine.orm.naming_strategy.underscore_number_aware  
        auto_mapping: true  
        mappings:  
            App:  
                type: attribute  
                is_bundle: false  
                dir: '%kernel.project_dir%/src/Entity'  
                prefix: 'App\Entity'  
                alias: App
```

DBAL (Database Abstraction Layer)

- Connexion MySQL via DATABASE_URL

ORM (Object-Relational Mapping)

- Mapping entités PHP
- Répertoire : src/Entity
- Namespace : App\Entity

Dans mon projet AutoRent

L'interaction avec la base de données repose sur Doctrine ORM, qui fait le pont entre les objets PHP et les tables SQL.

COMPOSANTS D'ACCÈS AUX DONNÉES - Entité (Mapping ORM)

Fichier : src/Entity/Reservation.php (partie 1)

```
namespace App\Entity;

use App\Repository\ReservationRepository;
use Doctrine\DBAL\Types\Types;
use Doctrine\ORM\Mapping as ORM;
use DateTimeImmutable;

#[ORM\Entity(repositoryClass: ReservationRepository::class)]
class Reservation
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column]
    private ?\DateTimeImmutable $dateReservation = null;
```

DÉCLARATION de mon entité

On importe des classes/bibliothèques pour pouvoir les utiliser dans notre code

Je MAPPE, Je fais CORRESPONDRE

Je RELIE PHP ↔ SQL

\$id en PHP correspond à id PRIMARY KEY en SQL

Le rôle du mapping est de donner des instructions à doctrine
Pour hydrater l'objet à partir des données de la base.

COMPOSANTS D'ACCÈS AUX DONNÉES - Entité (Relations)

Fichier : src/Entity/Reservation.php (partie 2)

Dans mon projet AutoRent, les relations ManyToOne créent les clés étrangères vers User et Vehicule.

```
#[ORM\ManyToOne(inversedBy: 'reservations')]  
private ?User $utilisateur = null;  
  
#[ORM\ManyToOne(inversedBy: 'reservations')]  
#[ORM\JoinColumn(nullable: false)]  
private ?Vehicule $vehicule = null;
```

`#[ORM\ManyToOne(inversedBy: 'reservations')]
private ?User $utilisateur = null;` Plusieurs réservations → Un seul utilisateur
Relation N:1

Crée **FK** : utilisateur_id → `user(id)` Relations ManyToOne - Clés étrangères

`inversedBy: 'reservations'` Permet navigation bidirectionnelle

jointure Elle se fait automatiquement grâce à symfony

COMPOSANTS D'ACCÈS AUX DONNÉES - Repository

Fichier : src/Repository/ReservationRepository.php

Dans mon projet AutoRent, le Repository contient les requêtes pour accéder aux réservations

```
?php  
  
namespace App\Repository;  
  
use App\Entity\Reservation;  
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;  
use Doctrine\Persistence\ManagerRegistry;  
  
/**  
 * @extends ServiceEntityRepository<Reservation>  
 */  
class ReservationRepository extends ServiceEntityRepository  
{  
    public function __construct(ManagerRegistry $registry)  
    {  
        parent::__construct($registry, Reservation::class);  
    }  
}
```

Le Repository est le composant qui permet de récupérer les données depuis la base.

Hérite de toutes les méthodes SANS LES ÉCRIRE (find, findAll, findBy, etc.) grâce à ServiceEntityRepository

COMPOSANTS D'ACCÈS AUX DONNÉES - Controller

Fichier : src/Repository/ReservationController.php

Dans mon projet AutoRent, le Controller utilise le repository et EntityManager pour orchestrer l'accès aux données

(Accès aux données)

```
namespace App\Controller
use App\Entity\Vehicule;
use App\Entity\Reservation;
use Doctrine\ORM\EntityManagerInterface;

final class ReservationController extends AbstractController {
#[Route('/reservation/{id}', name: 'app_reservation_new')]

public function new( Vehicule $vehicule,
EntityManagerInterface $entityManager
): Response
{
```

Le Controller

Il REÇOIT des données (lecture)

ParamConverter Doctrine vient d'une classe

Doctrine fait AUTOMATIQUEMENT :

1. Récupère {id} = 5 de l'URL
2. Fait : \$vehiculeRepository->find(5)
3. Hydrate l'objet Vehicule
4. L'injecte dans la méthode

L'EntityManager est le composant d'ÉCRITURE:
- persist() : Prépare INSERT
- flush() : Exécute les requêtes SQL

COMPOSANTS D'ACCÈS AUX DONNÉES - EntityManager

Composant d'écriture en base de données

Dans mon projet AutoRent, l'EntityManager gère toutes les opérations d'écriture : *INSERT*, *UPDATE* et *DELETE*.

```
$reservation->setUtilisateur($this->getUser());  
$reservation->setVehicule($vehicule);  
$reservation->setDateReservation(new  
DateTimelmmutable());  
$reservation->setStatut('en_attente');  
  
$vehicule->setStatut(false);
```

`$entityManager->persist($reservation);` *INSERT* Écriture en base
`$entityManager->persist($vehicule);` *UPDATE* Écriture en base
`$entityManager->flush();` *EXÉCUTE les requêtes SQL*

COMPOSANTS MÉTIER CÔTÉ SERVEUR

Fichier : src/Controller/ReservationController.php

Dans mon projet AutoRent, la logique métier implémente les règles spécifiques au domaine de la location : validations, calculs et services

VALIDATION MÉTIER

Vérifier la disponibilité

```
if (!$vehicule->isStatut()) {  
    $this->addFlash('error', 'Ce véhicule n\'est plus disponible.');//  
    return $this->redirectToRoute('app_vehicule_show', ['id' => $vehicule->getId()]);  
}
```

Re-vérifier juste avant sauvegarde

```
// Double vérification de disponibilité  
if (!$vehicule->isStatut()) {  
    $this->addFlash('error', 'Ce véhicule a été réservé entre temps.');//  
    return $this->redirectToRoute('app_vehicule_show', ['id' => $vehicule->getId()]);  
}
```

Calcul du prix

```
// Calculer le prix total  
$dateDebut = $reservation->getDateDebut();  
$dateFin = $reservation->getDateFin();  
$nombreJours = $dateDebut->diff($dateFin)->days + 1;//pour éviter de perdre un jour  
$prixTotal = $nombreJours * $vehicule->getTarifJournalier();  
$reservation->setPrixTotal($prixTotal);
```

\$vehicule->setStatut(false); Véhicule réservé = automatiquement indisponible

```
// Calculer le prix total  
$dateDebut = $reservation->getDateDebut();  
$dateFin = $reservation->getDateFin();  
$nombreJours = $dateDebut->diff($dateFin)->days + 1;//pour éviter de perdre un jour  
$prixTotal = $nombreJours * $vehicule->getTarifJournalier();  
$reservation->setPrixTotal($prixTotal);
```

Envoi emails admin + client

DÉPLOIEMENT DE L'APPLICATION

Mise en production du projet AutoRent

Déploiement de l'application AutoRent en environnement local sécurisé

Avant de procéder au déploiement de l'application AutoRent, j'ai effectué une phase de tests et de validation du code pour garantir la qualité et la fiabilité de l'application. Validation du code avec le W3C Validator



The screenshot shows a code editor with a sidebar on the left displaying a file tree. The tree includes a 'tests' folder containing 'Functional' and 'Unit' subfolders. Under 'Functional', several files are listed: 'VehiculeControllerTest.php', 'CategorieControllerTest.php', 'ContactControllerTest.php', 'HomeControllerTest.php', 'RegistrationControllerTest.php', 'ReservationControllerTest.php', 'SecurityControllerTest.php', and 'VehiculeControllerTest.php'. The 'VehiculeControllerTest.php' file is currently selected and its content is displayed in the main pane. The code is written in PHP and includes comments and several public functions.

```
14 }  
15  
16 // Test si la méthode vehicule() existe  
17 public function testVehiculeMethodExists()  
18 {  
19     $this->assertTrue(method_exists(VehiculeController::class, 'vehicule'));  
20 }  
21  
22 // Test si la méthode vehicule() est publique  
23 public function testVehiculeMethodIsPublic()  
24 {  
25     $reflection = new \ReflectionMethod(VehiculeController::class, 'vehicule');  
26     $this->assertTrue($reflection->isPublic());
```

Pour le déploiement de l'application AutoRent en production, j'ai sélectionné **AlwaysData**, un hébergeur web français:

Support **Symfony** et PHP 8
Base de données **MySQL**
Accès **FTP**.

The screenshot shows the 'Notifications' section of the alwaysdata web interface. The top navigation bar includes links for Home, Notifications, WARNINGS, ACTIONS, and SYSTEM TASKS. A sidebar on the left contains dropdown menus for location-voiture (Disk, Web, Domains), Disk, Web, and Domains. The main content area displays a single notification entry:

Date	Information
Oct 3, 2025, 9:18:45 AM	[alwaysdata] Welcome to alwaysdata!

Utilisation de FileZilla pour le transfert, de l'application Autorent vers le serveur

AlwaysData :

Fichiers transférés : Code source complet

Côté serveur distant

Configuration du fichier .env en mode production :

APP_ENV=prod

Connexion MySQL configurée :

DATABASE_URL="mysql://436533:9326yez10@mysql-location-voiture.alwaysdata.net
:3306/location-voiture_rent?serverVersion=8.0.32&charset=utf8mb4"

Nom de fichier	Taille d...	Type de ...	Dernière m...	Droits d'...	Propriéta...
..					
.phpunit.cache		Dossier ...	20/10/2025...	flcdmpe ...	location-...
api		Dossier ...	20/10/2025...	flcdmpe ...	location-...
assets		Dossier ...	20/10/2025...	flcdmpe ...	location-...
bin		Dossier ...	20/10/2025...	flcdmpe ...	location-...
config		Dossier ...	20/10/2025...	flcdmpe ...	location-...
migrations		Dossier ...	20/10/2025...	flcdmpe ...	location-...
public		Dossier ...	20/10/2025...	flcdmpe ...	location-...
src		Dossier ...	20/10/2025...	flcdmpe ...	location-...
templates		Dossier ...	20/10/2025...	flcdmpe ...	location-...
tests		Dossier ...	20/10/2025...	flcdmpe ...	location-...
translations		Dossier ...	20/10/2025...	flcdmpe ...	location-...
var		Dossier ...	20/10/2025...	flcdmpe ...	location-...
vendor		Dossier ...	20/10/2025...	flcdmpe ...	location-...
.env	2210	Fichier E...	20/10/2025...	adfrw (0...)	location-...
.env.dev	0	Fichier D...	20/10/2025...	adfrw (0...)	location-...
.env.test	197	Fichier T...	20/10/2025...	adfrw (0...)	location-...
.gitignore	429	Fichier s...	20/10/2025...	adfrw (0...)	location-...
compose.override...	349	Fichier s...	20/10/2025...	adfrw (0...)	location-...

Création des tables de la base de données:

MySQL host: mysql-location-voiture.alwaysdata.net
Version: 10.11 (mariadb)

[phpMyAdmin](#)

DATABASES | USERS | PROCESSES

Name

location-voiture_rent

Connexion SSH au serveur
AlwaysData .

Depuis le SSH je me mets de dossier
du WWW .

J'installe les dépendances avec
`composer install`.

Création des tables de la base de
données avec
`doctrine:migrations:migrate`

Compilation des asset - map:compile

J'ai tapé `php bin/console cache:clear --env=prod`. Le cache
s'est vidé

Installation de dépendances:

je vais utiliser les commandes de mon terminal ssh pour en y installer les dépendances.

Je vais executer les migrations

php bin/console doctrine:migration:migrate

Je vais compiler les asset map

php bin/console asset-map:compile --env=prod

Je vais vider le cache

php bin/console cache:clear --env=prod

Je configure depuis alwaysdata point d'entrer à al racine

Répertoire racine *

/accueil/location-voiture/

www/public

Configuration Apache avec Symfony Apache Pack

toujours depuis la commande de mon terminal
c'est le **Bundle de symfony**
composer require symfony/apache-pack
URLs "propres" et SEO

Résultat obtenu au final dans la barre d'url



Cela a permis de faire fonctionner le site sur toutes les url

Vous pouvez visiter le site sur : <https://location-voiture.alwaysdata.net>