

TITRE PROFESSIONNEL DÉVELOPPEUR WEB ET WEB MOBILE

Soutenance

David CHROSCIEL-ODEGAARD– MyScript

Qui suis-je?

Parcours

-Développeur full stack, 2020-2021

ICS Nice

-Game design, 2021-2023

Artline

-Développeur full stack, 2024-2025

La Plateforme_

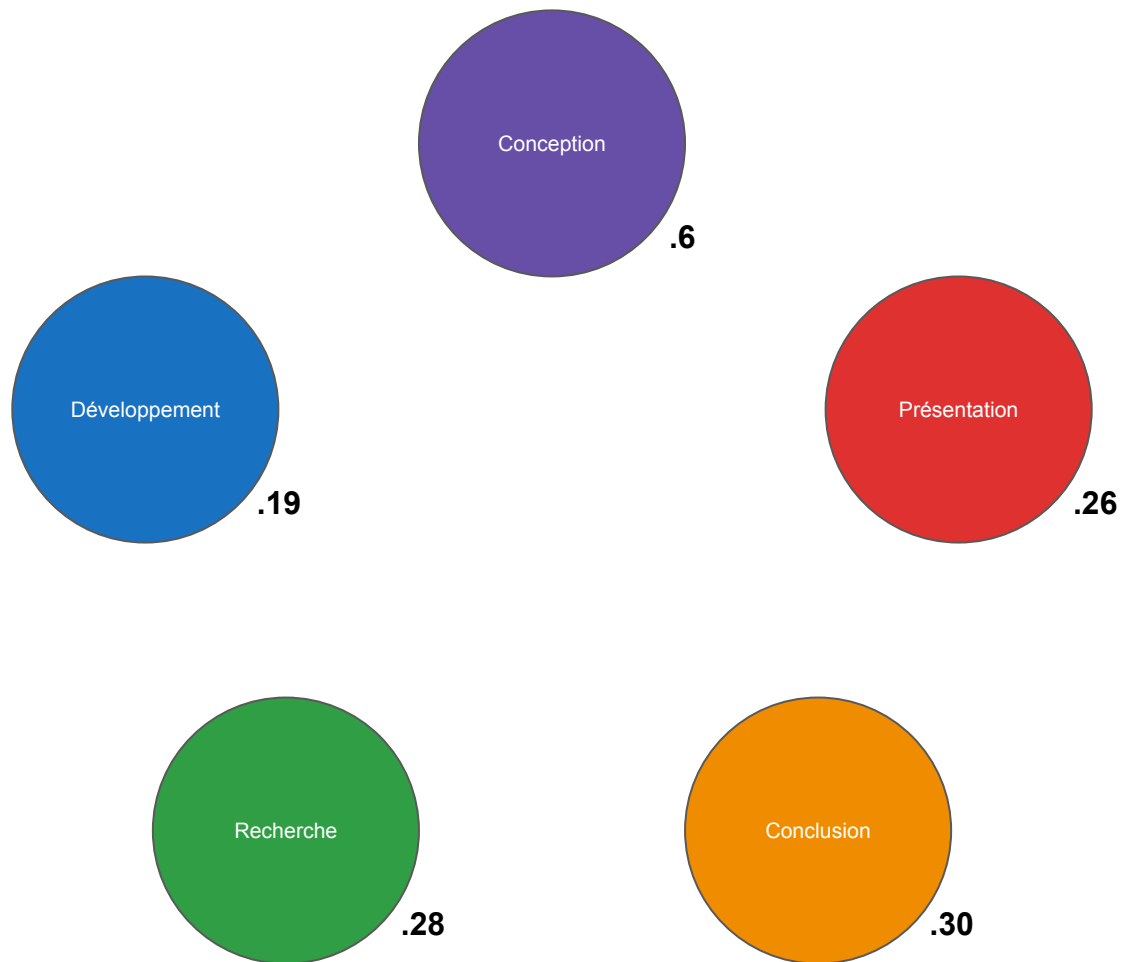
Pourquoi

-Passion depuis mon enfance pour les nouvelles technologies

-Le développement pour mieux comprendre la logique applicative

-Approfondir mes compétences techniques en développement web

Sommaire



Index

1. **Titre de page**
2. Qui suis-je ?
3. **Sommaire**
4. "Vous êtes ici"
5. Pourquoi MyScript ?
6. **Conception**
7. User Stories 1/2
8. User Stories 2/2
9. Arborescence 1/2
10. Arborescence 2/2
11. Wireframe
12. Maquettes
13. MCD
14. MLD
15. Dictionnaire des données 1/2
16. Dictionnaire des données 2/2
17. Stack technique
18. Gitflow
19. **Développement**
20. Architecture
21. Création de la base de données
22. Accès aux données
23. Réalisation du back-end
24. Réalisation du front-end
25. Sécurité
26. **Instant présentation**
27. Réalisation figma
28. **Exemple de recherche**
29. SQLite contre MySQL
30. **Conclusion**

Pourquoi MyScript ?

J'ai constaté

Un code utile pour tout le monde ne rapporte
parfois aucun revenus à son créateur

(Soutien?)

Audience

- Habitué de l'informatique
- Adeptes du code

Une organisation de travail ?

Conception

User Stories 1/2

- 3 types d'utilisateurs

- Utilisateur
- Vendeur
- Administrateur

vert : mvp

mvp = "minimum viable product"

en Tant que	Je souhaite	Afin de
utilisateur/vendeur	m'enregistrer à l'appli	bénéficier aux différents services
utilisateur/vendeur	me connecter à l'appli	accéder aux différents services
utilisateur/vendeur	me déconnecter de l'appli	protéger mes informations personnelles
utilisateur/vendeur/administrateur	accéder à mes informations	modifier mes informations personnelles
utilisateur/vendeur/administrateur	personnaliser mes préférences	avoir une appli confortable
utilisateur	avoir un panier	placer mes achats
utilisateur	avoir des souhaits	faire des achats futurs
utilisateur	avoir de l'aide	trouver de l'assistance
utilisateur	valider mon achat	profiter de mon achat
vendeur	accéder à mes articles	gérer mes articles
administrateur	accéder à la liste des utilisateurs	gérer les profils d'utilisateurs
administrateur	accéder aux logs des utilisateurs	modérer les utilisateurs
administrateur	accéder à la liste des articles	gérer les articles
utilisateur/vendeur/administrateur	accéder à la liste des articles	voir les articles

User Stories 2/2

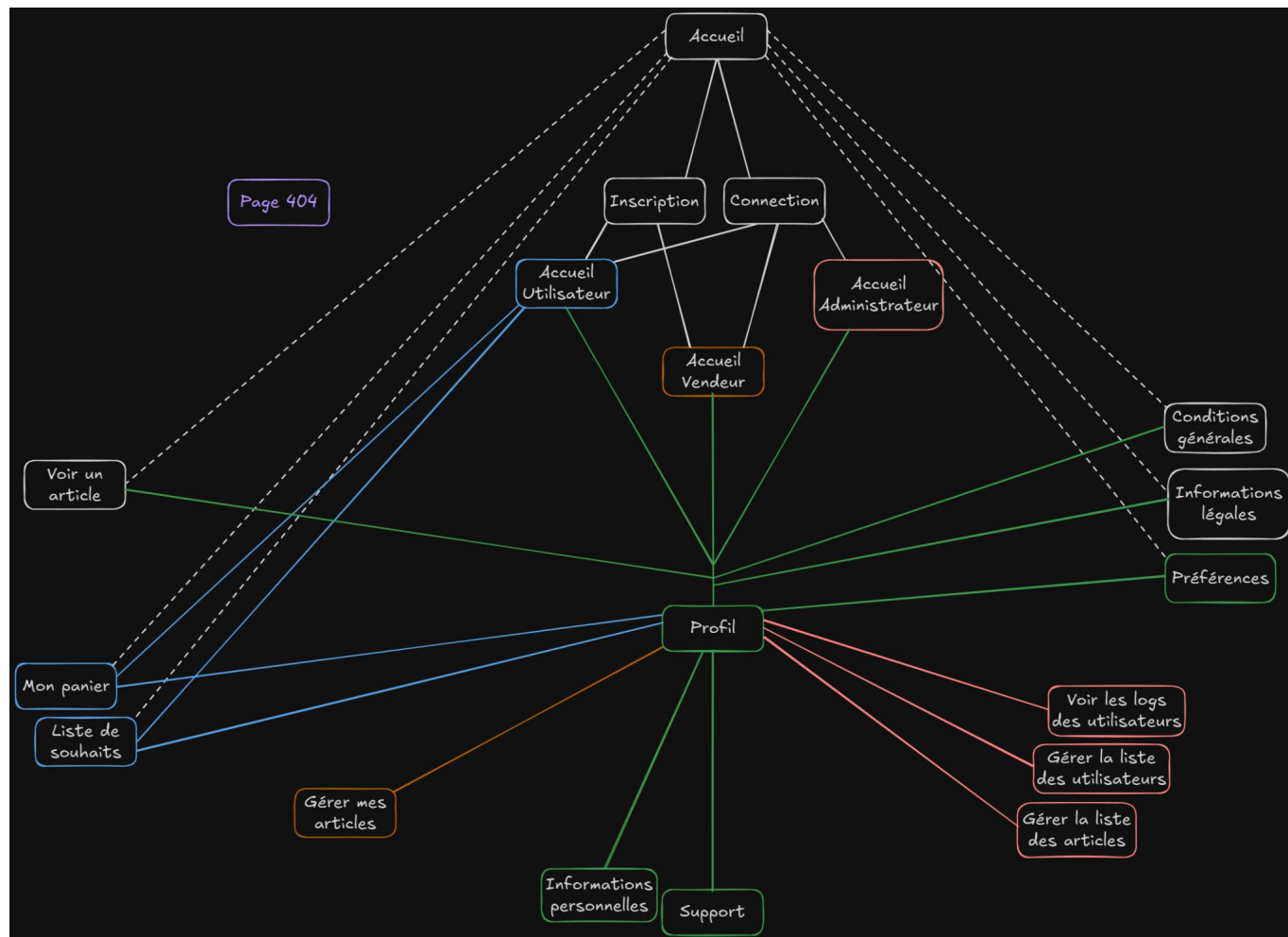
en Tant que	Je souhaite	Afin de
utilisateur	avoir un historique d'achat	une preuve de mon achat
vendeur	recevoir mes bénéfices	rentabiliser mon article
utilisateur/vendeur/administrateur	filtrer mes recherches d'articles	trouver mes attentes
utilisateur/vendeur/administrateur	personnaliser mes préférences	avoir une appli confortable
utilisateur/vendeur/administrateur	avoir un espace "messages"	communiquer avec les vendeurs/administrateurs

Jaune : futur mvp

Arborescence

(mvp)

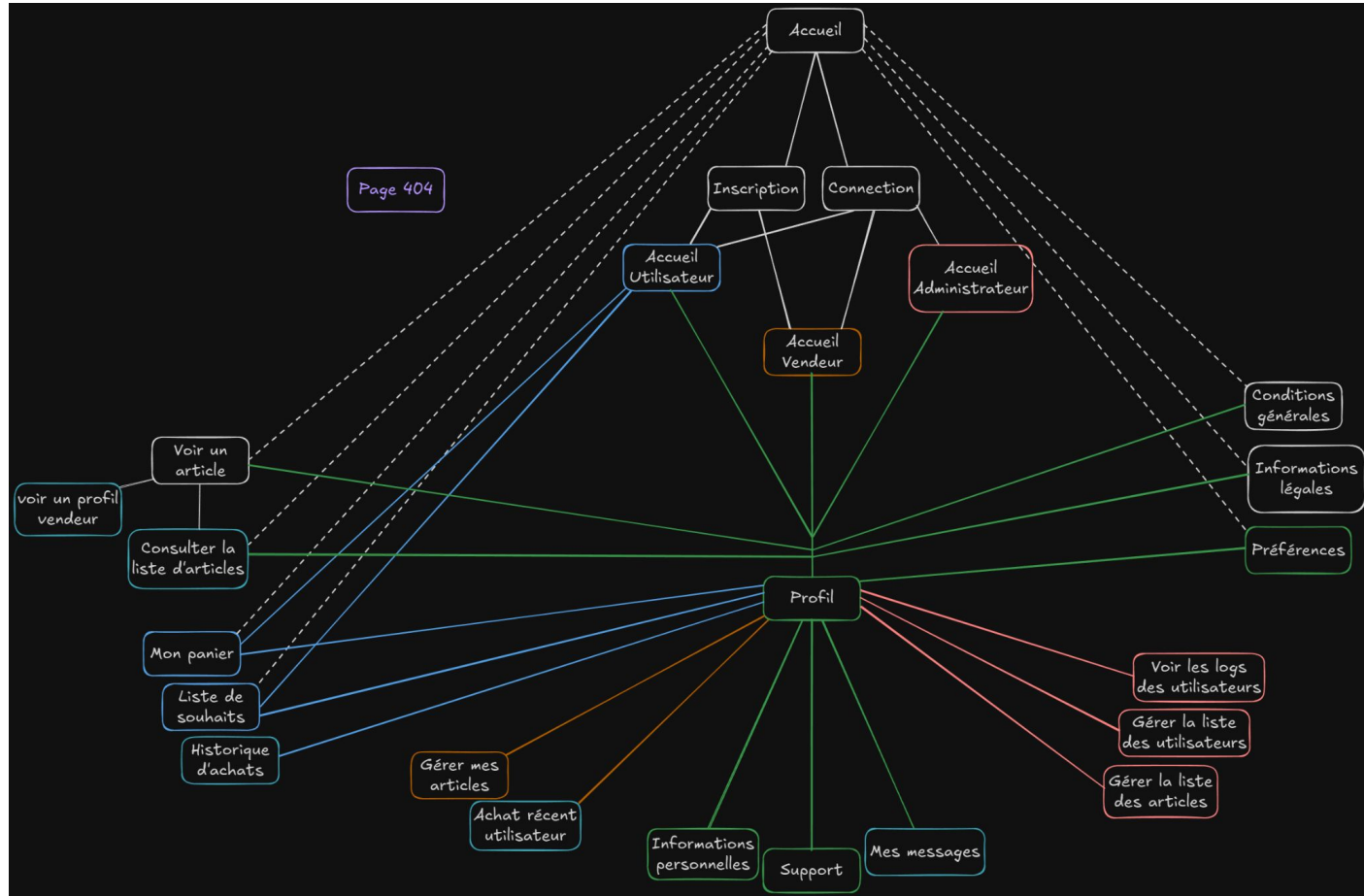
- - Everyone
- - Utilisateur
- - Vendeur
- - Administrateur
- - Accès commun pour utilisateur/vendeur/administrateur



Arborescence

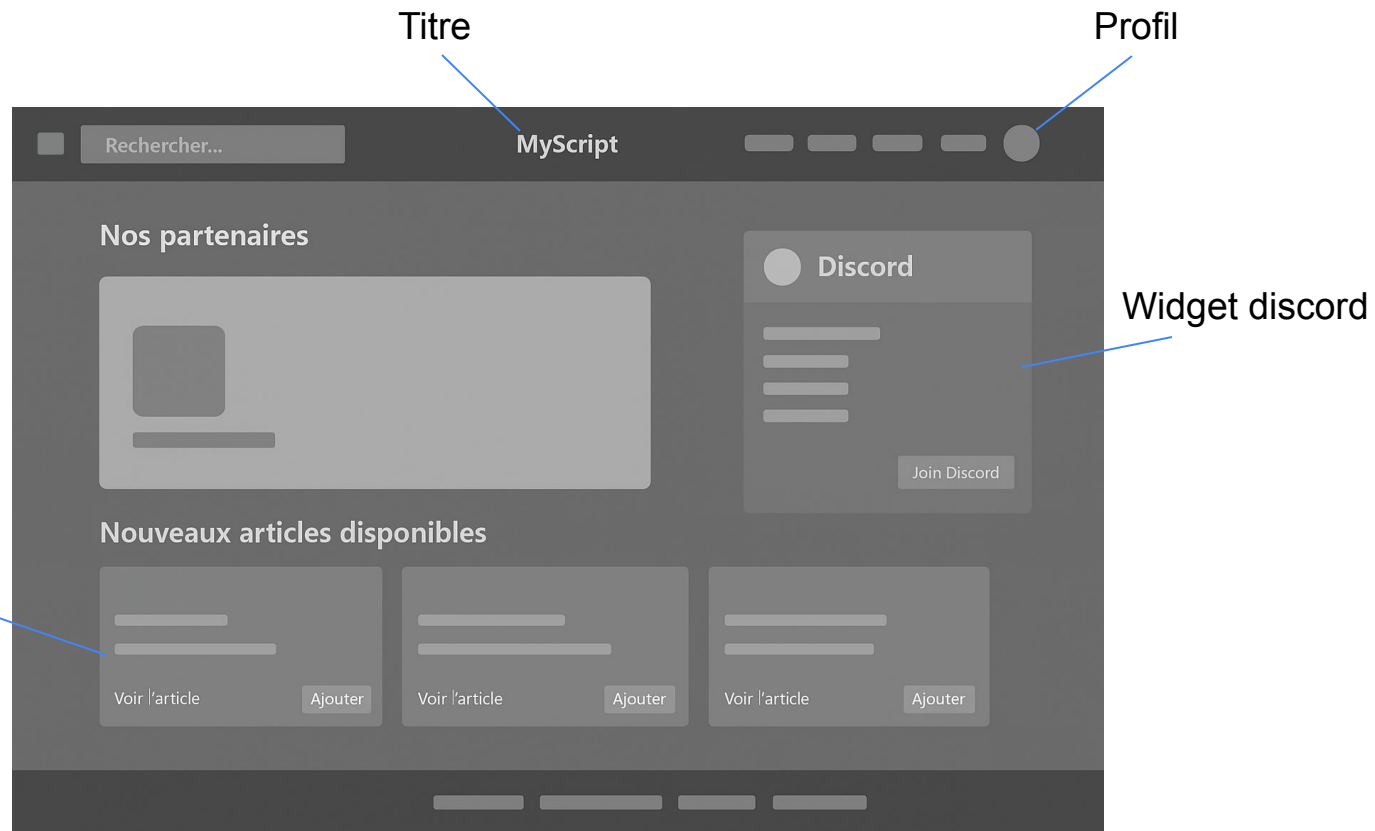
(futur mvp)

● - Futur mvp

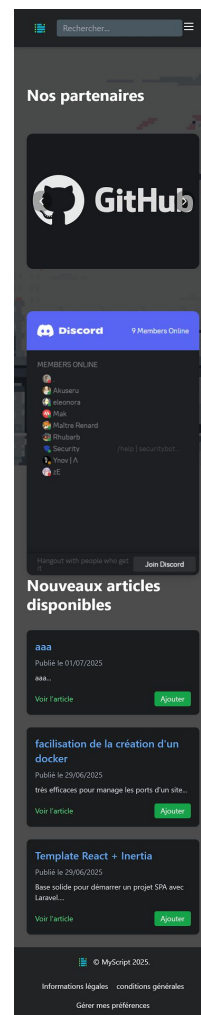
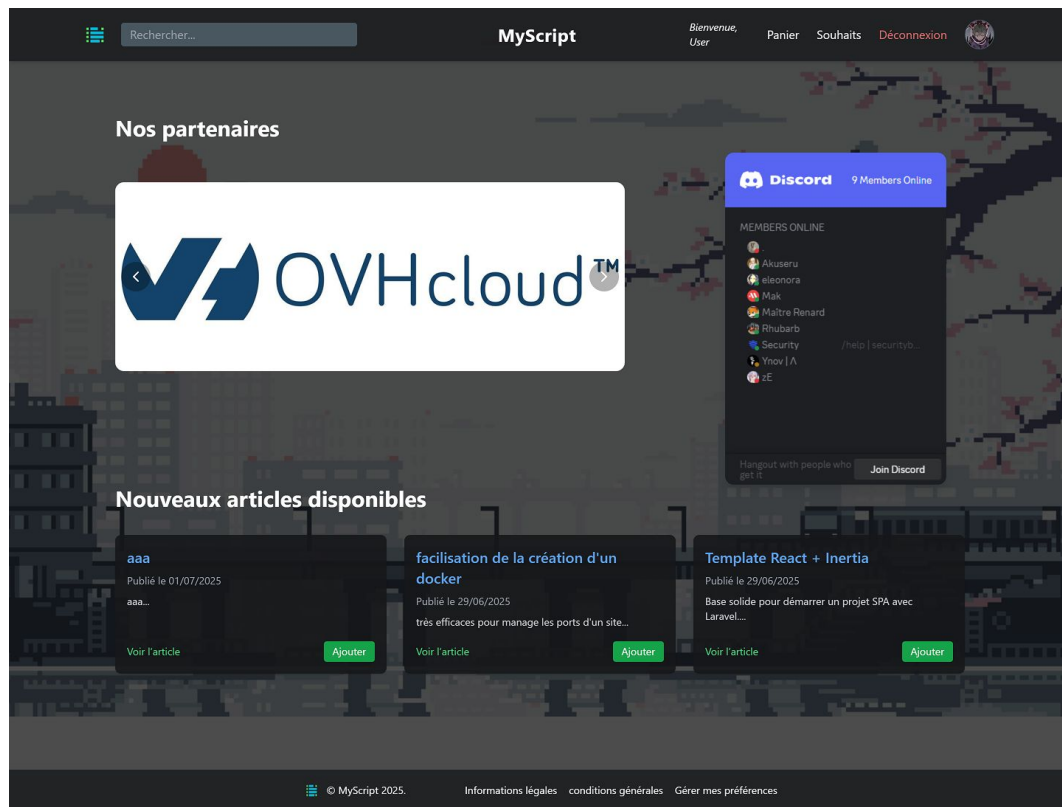


Wireframe

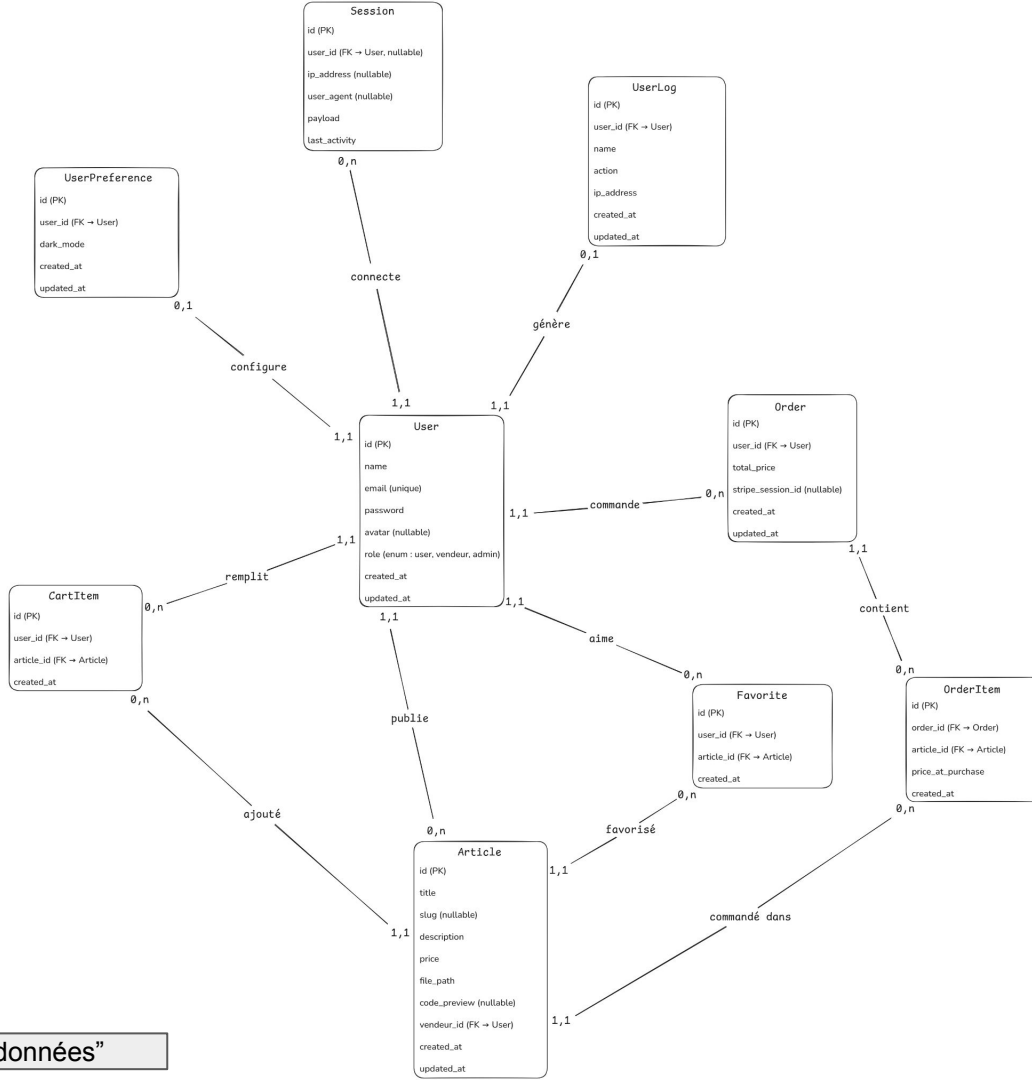
Redirection page article



Maquettes



MCD



mcd = "modèle conceptuel de données"

MLD

user (id, name, email, password, avatar, role, created_at, updated_at)

article (id, title, slug, description, price, file_path, code_preview, created_at, updated_at, #vendeur_id)

order (id, total_price, stripe_session_id, created_at, updated_at, #user_id)

order_item (id, price_at_purchase, created_at, #order_id, #article_id)

cart_item (id, created_at, #user_id, #article_id)

favorite (id, created_at, #user_id, #article_id)

user_preference (id, dark_mode, created_at, updated_at, #user_id)

user_log (id, name, action, ip_address, created_at, updated_at, #user_id)

session (id, ip_address, user_agent, payload, last_activity, #user_id nullable)

Dictionnaire des données 1/2

User

Nom de la donnée	Désignation	Type	Contrainte
id	Identifiant unique de l'utilisateur	integer	generated as identity primary key
name	Nom de l'utilisateur	text	not null
email	Adresse email de l'utilisateur	text	not null, unique
password	Mot de passe de l'utilisateur	text	not null
avatar	URL de l'avatar de l'utilisateur	text	nullable
role	Rôle de l'utilisateur	enum	check("user", "vendeur", "admin")
created_at	Date de création de l'utilisateur	timestamp	not null
updated_at	Date de mise à jour de l'utilisateur	timestamp	not null

Dictionnaire des données 2/2

Article

Nom de la donnée	Désignation	Type	Contrainte
id	Identifiant unique de l'article	integer	generated as identity primary key
title	Titre de l'article	text	not null
slug	Slug de l'article	text	nullable
description	Description de l'article	text	not null
price	Prix de l'article	decimal	not null
file_path	Chemin du fichier de l'article	text	not null
code_preview	Aperçu du code de l'article	text	nullable
vendeur_id	Identifiant du vendeur	integer	foreign key user_id

Stack technique

FRONT	BACK	DEVELOPER EXPERIENCE
<ul style="list-style-type: none">-React (via Inertia.js)-Tailwind CSS-Vite-Lucide React	<ul style="list-style-type: none">-Laravel-MySQL-Stripe	<ul style="list-style-type: none">-Git/GitHub-phpMyAdmin-Thunder Client-artisan-npm

Gitflow

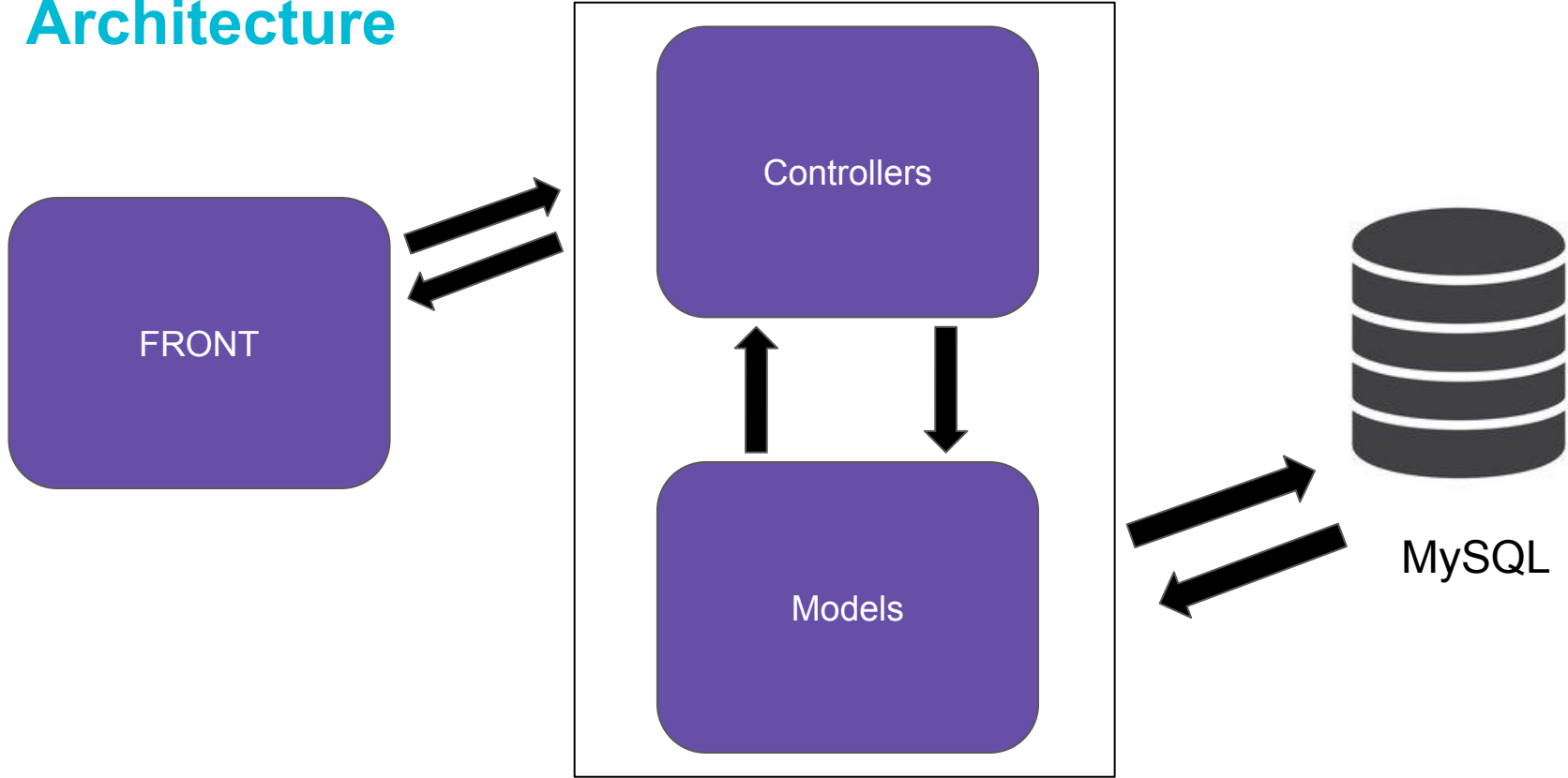


MAIN : Branche incluant les versions fonctionnelles et stables

DEV : Branche de production

Développement

Architecture



Création de la base de données

```
$ php artisan make:migration create_users_table
```

```
1 php artisan migrate
```

```
1 php artisan migrate:fresh
```

```
2
```

```
3 php artisan migrate:fresh --seed
```

▼ migrations

🐘 2025_04_25_003657_create_sessions_table.php

🐘 2025_05_06_073427_create_users_table.php


🐘 2025_05_06_074006_create_articles_table.php

🐘 2025_05_06_075436_create_orders_table.php

🐘 2025_05_06_075657_create_order_items_table.php

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 Lam-Area, 2 months ago | 1 author (Lam-Area)
8 return new class extends Migration {
9     public function up(): void
10     {
11         Schema::create('users', function (Blueprint $table) {
12             $table->id();
13             $table->string('name', 100);
14             $table->string('email', 150)->unique();
15             $table->string('password', 255);
16             $table->enum('role', ['user', 'vendeur', 'admin']);
17             $table->dateTime('created_at');
18             $table->dateTime('updated_at');
19         });
20     }
21
22     public function down(): void
23     {
24         Schema::dropIfExists('users');
25     }
26 };
27
```

Accès aux données


Bienvenue dans phpMyAdmin

Langue (Language)
Français - French

Connexion ⓘ

Utilisateur : root

Mot de passe :

Choix du serveur : MySQL

Connexion

phpMyAdmin

Serveur courant : MySQL

Récentes Préférences

Nouvelle base de données

- myscript
 - Nouvelle table
 - articles
 - cache
 - cache_locks
 - cart_items
 - favorites
 - migrations
 - orders
 - order_items
 - sessions
 - users
 - user_logs
 - user_preferences

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations Déclencheurs

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0.0001 seconde(s))

`SELECT * FROM `users``

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

Options supplémentaires

		id	name	email	password	avatar	role	created_at	updated_at
<input type="checkbox"/>	Éditer Copier Supprimer	1	Admin Test	admin@myscript.com	\$2y\$12\$sk0S37la8XeL3eKmwJBbyenJOv8ZWJAd9mZ5Um4DNXA...	NULL	admin	2025-06-29 15:17:57	2025-06-29 15:17:57
<input type="checkbox"/>	Éditer Copier Supprimer	2	Vendeur Test	vendeur@myscript.com	\$2y\$12\$8NwhQLIMx9nz2bqTtiA7ie36MPZCQMz3VtoSuiO0ouA...	avatars/92u2o8w4TEldIPOD1X4VMg7qo08RHUYZ2DVO29IZ...	vendeur	2025-06-29 15:17:57	2025-07-01 01:35:26
<input type="checkbox"/>	Éditer Copier Supprimer	3	User	user@myscript.com	\$2y\$12\$mVcLTJB84Brk1WO5pLEyedj7TicgmsVIT6mU1hdbD9...	avatars/BDBlnMvhuGdFgW95QrRleb3MOCP4ak03m4SSATH0...	user	2025-06-29 15:17:57	2025-07-01 01:39:34
<input type="checkbox"/>	Éditer Copier Supprimer	4	aaa	aaa@myscript.com	\$2y\$12\$UxqCv6E1ThJ4VWczlzChOoH0/9k2KezXHUmxMNV9k...	avatars/z9Nxa2ai8lbpkipXwQ8E8YICHdza4EICNYEDs2CX.g...	vendeur	2025-07-01 01:08:21	2025-07-01 01:36:02

☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

☐ Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

Opérations sur les résultats de la requête

Imprimer Copier dans le presse-papiers Exporter Afficher le graphique Créer une vue

Réalisation du back-end

ROUTES

web.php

MIDDLEWARES

HandleInertia

CONTROLLERS

MODELS

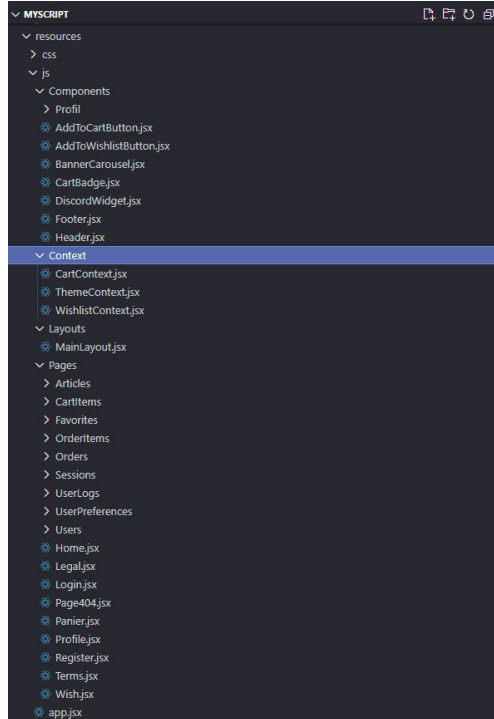
```
routes / web.php /...
37 });
38
39 /* auth */
40 Route::post('/register', [AuthController::class, 'register']);
41 Route::get('/login', [AuthController::class, 'showLogin'])->name('login');
42 Route::post('/login', [AuthController::class, 'login']);
43
44 /* profil */
45 Route::middleware('auth')->group(function () {
46     Route::get('/profile', function () {
47         $user = Auth::user();
48         $articles = Article::where('vendeur_id', $user->id)->get();
49
50         $users = $user->role === 'admin'
51             ? User::select('id', 'name', 'email', 'role')->get()
52             : [];
53
54         $logs = $user->role === 'admin'
55             ? UserLog::with('user')->latest()->get()
56             : [];
57
58         return Inertia::render('Profile', [
59             'articles' => $articles,
60             'users' => $users,
61             'logs' => $logs,
62             'auth' => [
63                 'user' => $user,
64                 'isAdmin' => $user->role === 'admin',
65                 'isVendeur' => $user->role === 'vendeur',
66                 'isUser' => $user->role === 'user',
67             ],
68         ]);
69     });
70
71     Route::put('/profile/info', [UserController::class, 'updateInfo'])->name('profile.update');
72     Route::post('/logout', [AuthController::class, 'logout']);
73 });
74
```

```
pp > Http > Middleware > HandleInertiaRequests.php > ...
9 class HandleInertiaRequests extends Middleware
10 {
24 }
25
26 /**
27  * props share par défaut
28  */
29 public function share(Request $request): array
30 {
31     return array_merge(parent::share($request), [
32         'auth' => [
33             'user' => fn () => $request->user(),
34             'role' => fn () => $request->user()->role,
35             'isUser' => fn () => $request->user()->role === 'user',
36             'isAdmin' => fn () => $request->user()->role === 'admin',
37             'isVendeur' => fn () => $request->user()->role === 'vendeur',
38         ],
39         'userPreference' => fn () => $request->user()
40             ? UserPreference::where('user_id', $request->user()->id)->first()
41             : null,
42     ]);
43 }
44 }
45
```

```
11 class AuthController extends Controller
12 {
13     public function register(Request $request)
14     {
15         $validated = $request->validate([
16             'name' => 'required|string|max:100',
17             'email' => 'required|email|unique:users,email',
18             'password' => 'required|string|min:6|confirmed',
19             'role' => 'required|in:user,vendeur',
20         ]);
21
22         $user = \App\Models\User::create([
23             'name' => $validated['name'],
24             'email' => $validated['email'],
25             'password' => bcrypt($validated['password']),
26             'role' => $validated['role'],
27             'created_at' => now(),
28             'updated_at' => now(),
29         ]);
30     }
31 }
```

ex : panier.jsx

Réalisation du frontend



Appli découper en différents types de dossiers :

- composants
- context
- layouts
- pages

```
1 import { useCart } from '@context/CartContext';
2 import { usePage, Link } from '@inertiasjs/react';
3 import MainLayout from '@Layouts/MainLayout';
4
5 export default function Panier() {
6   const { cart, removeFromCart } = useCart();
7   const { auth } = usePage().props;
8
9   const handleCheckout = () => {
10     if (!auth.user) {
11       window.location.href = '/login';
12     } else {
13       alert('Paiement initié avec ' + cart.length + ' article(s)');
14     }
15   };
16
17   return (
18     <MainLayout>
19       <div className="flex flex-col min-h-screen text-white">
20         <div className="flex-1 px-6 py-10">
21           <h1 className="text-2xl font-bold mb-6">Votre panier</h1>
22
23           {cart.length === 0 ? (
24             <p className="text-gray-400">Aucun article dans le panier.</p>
25           ) : (
26             <ul className="space-y-4">
27               {cart.map((item) => (
28                 <li key={item.id} className="border p-4 rounded shadow bg-[#16171A] bg-opacity-90">
29                   <div className="flex justify-between items-center">
30                     <div>
31                       <h2 className="font-semibold">{item.title}</h2>
32                       <p className="text-sm text-gray-400">{item.price} €</p>
33                     </div>
34                     <div className="flex gap-4 items-center">
35                       <Link
36                         href={`/articles/${item.id}`}
37                         className="text-green-400 hover:underline">
38                         Voir l'article
39                       </Link>
40                       <button
41                         onClick={() => removeFromCart(item.id)}
42                         className="text-red-500 hover:underline">
43                         Supprimer
44                       </button>
45                     </div>
46                   </div>
47                 </li>
48               ))}
49             </ul>
50
51             <div className="mt-6 flex justify-end">
52               <button
53                 onClick={handleCheckout}
54                 className="bg-blue-600 text-white px-6 py-2 rounded hover:bg-blue-700">
55                 Valider l'achat
56               </button>
57             </div>
58           )}
59         </div>
60       </MainLayout>
61     );
62   }
63 }
```


Sécurité

FAIT	A FAIRE
<ul style="list-style-type: none">- validation des données- mdp hashés (bcrypt)- authentification sécurisée- prévention contre injection sql (eloquent)- middleware (protection des routes)	<ul style="list-style-type: none">- suppression des pages inutiles- meilleure protection des routes- captcha- connection par double authentification- connection via google- faille xss

Instant présentation

Réalisation figma



Exemple de recherche

SQLite contre MySQL

hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference



Avantages et inconvénients de SQLite par rapport à MySQL

Résumons rapidement les différences essentielles entre les deux options ci-dessous.

Avantages de SQLite :

- Basé sur des fichiers et facile à configurer et à utiliser
- Convient au développement et aux tests de base
- Facilement portable
- Utilise la syntaxe SQL standard avec des modifications mineures
- Facile à utiliser

Inconvénients de SQLite :

- Manque de fonctionnalités de gestion des utilisateurs et de sécurité
- Pas facilement évolutif
- Ne convient pas aux grandes bases de données
- Ne peut pas être personnalisé

Avantages de MySQL :

- Facile à utiliser
- Fournit de nombreuses fonctionnalités liées à la base de données
- Bonnes fonctionnalités de sécurité
- Facilement évolutif et convient aux grandes bases de données
- Offre une bonne vitesse et de bonnes performances
- Offre une bonne gestion des utilisateurs et de multiples contrôles d'accès

Inconvénients de MySQL :

- Nécessite une certaine expertise technique pour la configuration
- Syntaxe légèrement différente par rapport au SQL conventionnel

Conclusion

Satisfaction du travail accompli :
projet fonctionnel et presque complet

Montée en connaissance grâce à ce
projet

Approfondissement sur mes points
forts et mes axes d'amélioration

Merci !

Je suis à votre écoute !