

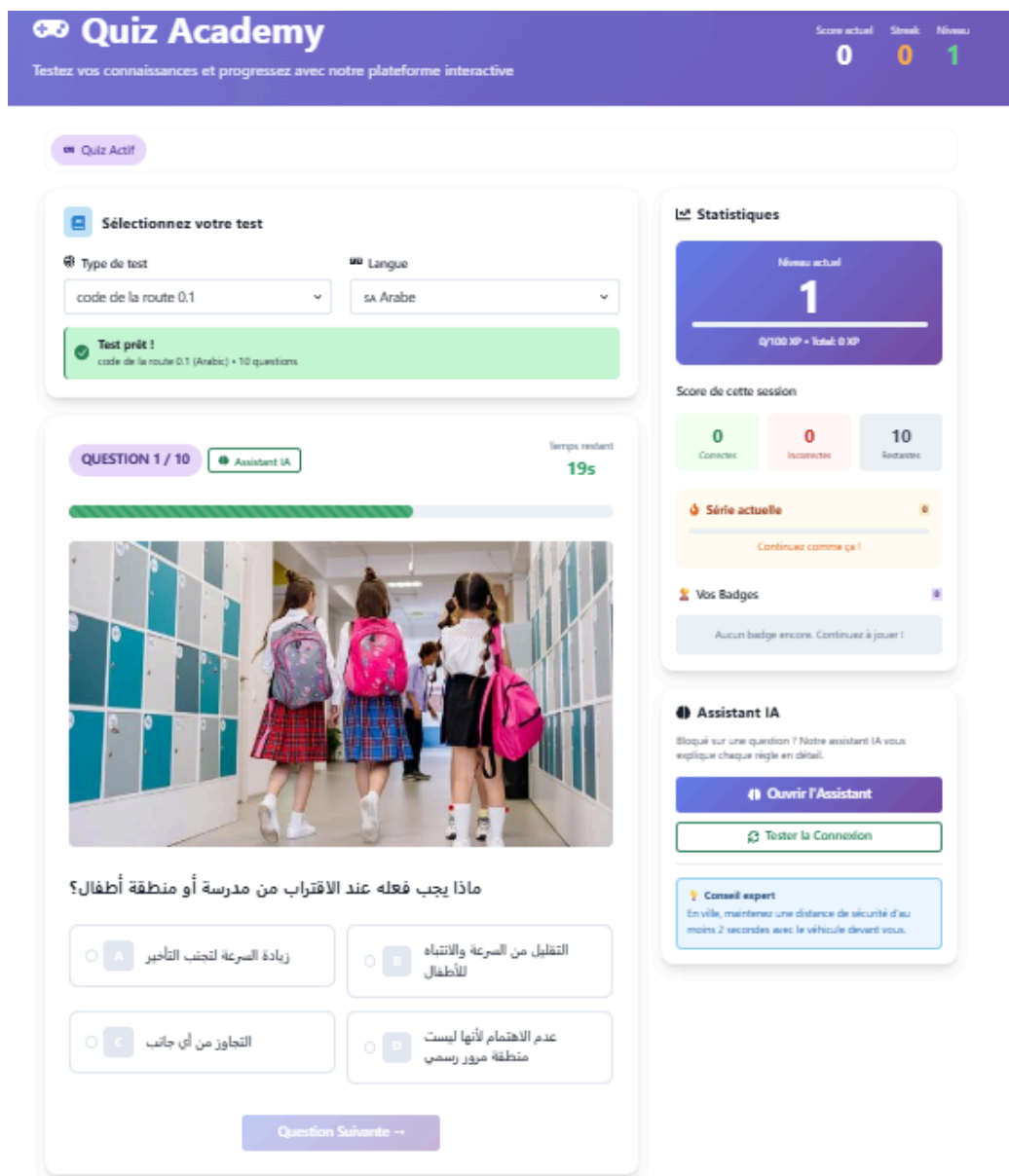
# Projet : Quiz Hub - Application MERN avec IA Intégrée

Wiem Hemdi | 5 DS 1

## 1. Présentation Générale du Projet

### 1.1 Objectif Principal

Créer une plateforme d'apprentissage interactive du code de la route qui transforme la formation théorique en une expérience engageante, motivante et personnalisée grâce aux technologies modernes et à l'intelligence artificielle.



## Points clés visibles :

- Design moderne avec Chakra UI
- Interface bilingue (Arabe/Français)
- Éléments visuels engageants
- Navigation intuitive

## 2.Fonctionnalités Clés

### 2.1. Système de Quiz Intelligent

- ✓ Questions aléatoires et adaptatives
- ✓ Timer par question (20 secondes)
- ✓ Feedback immédiat avec sons
- ✓ Correction détaillée
- ✓ Progression en temps réel

### 2.2. Assistant IA Personnel 🤖

**Maîtrisez le code de la route avec intelligence**

La plateforme d'apprentissage la plus avancée, combinant IA, quiz interactifs et suivi personnalisé pour votre réussite.

Commencer un Quiz | Assistant IA | Mon Progress

10K+ Apprenants | 95% Réussite | 500+ Questions | 24/7 Support

**Assistant IA**  
Expert en code de la route [Effacer]

**\*\*EXPLICATION:\*\*** Le feu orange annonce le rouge. Freinez si vous pouvez le faire sans danger.

**\*\*SANCTION:\*\*** Franchissement = considéré comme feu rouge

**\*Référence:** Article R412-31\*

expert-database

Alcool au volant : quelle limite ?

**\*\*Code de la Route - Expert\*\***

**\*\*RÈGLE:\*\*** 0,5 g/L maximum (0,2 g/L jeunes)

**\*\*EXPLICATION:\*\*** Taux sanguin maximum. Jeunes conducteurs (<3 ans): 0,2 g/L.

**\*\*SANCTION:\*\*** 0,5-0,8 g/L: 135€ + 6 points

**\*Référence:** Article L234-1\*

expert-database

**Questions rapides :**


Qu'est-ce qu'un feu orange ? | Distance de sécurité minimale ?




Comment prendre un rond-point ? | Alcool au volant : quelle limite ?

Posez votre question sur le code de la route...

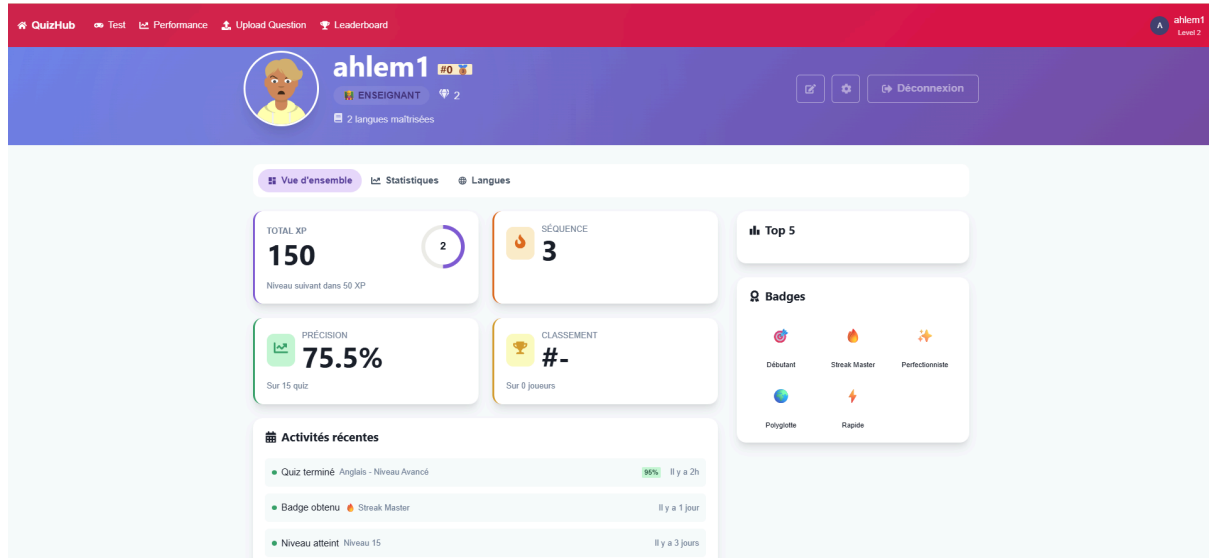
Assistant IA spécialisé [Envoyer]


## Avantages uniques :


-  Gratuit - Pas d'abonnement API

-  Confidentiel - Données restent locales
-  Rapide - Réponses instantanées
-  Précis - Spécialisé code de la route


## 2.3. Système de Gamification Avancé 🏆



 **XP Dynamique :** 10-50 points/question

 **Streak Bonus :** Multiplicateurs pour série

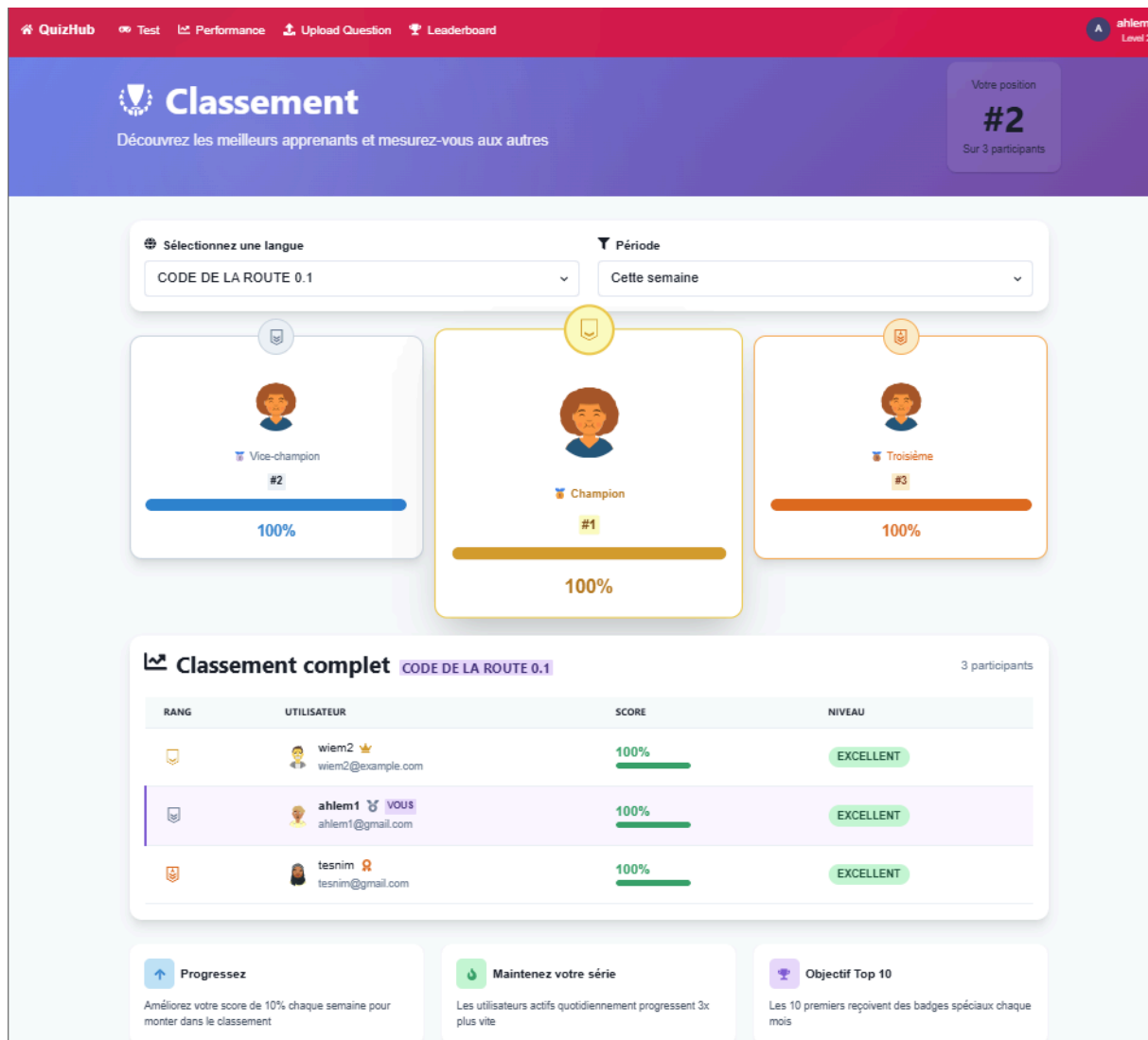
 **Badges :** 10+ badges à collectionner

 **Niveaux :** 1 à 100+ niveaux de progression

 **Leaderboards :** Classements locaux/nationaux

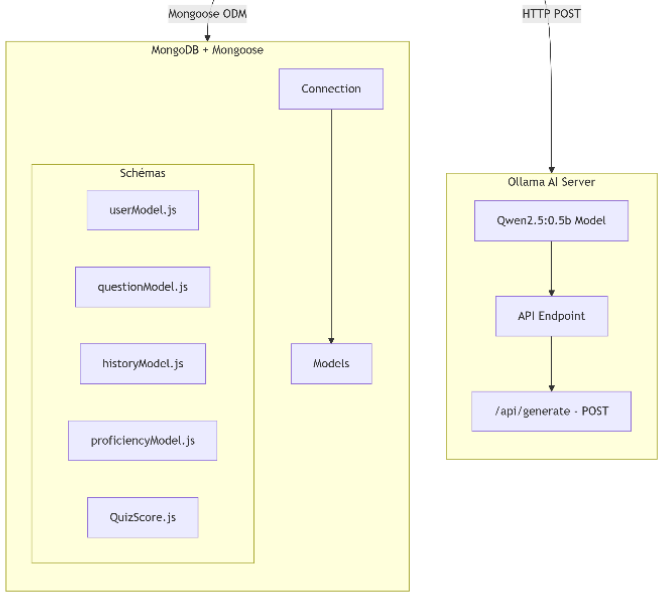
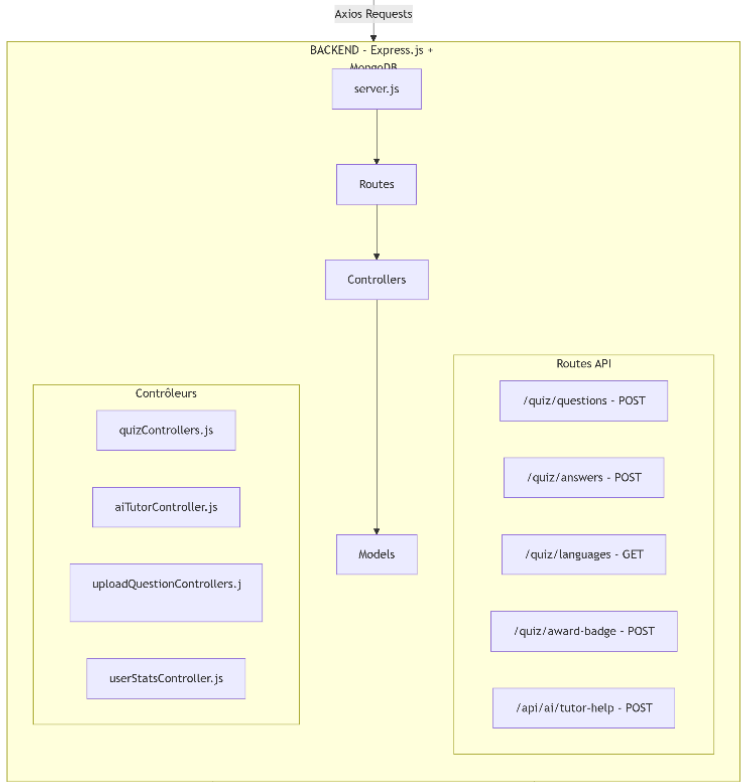
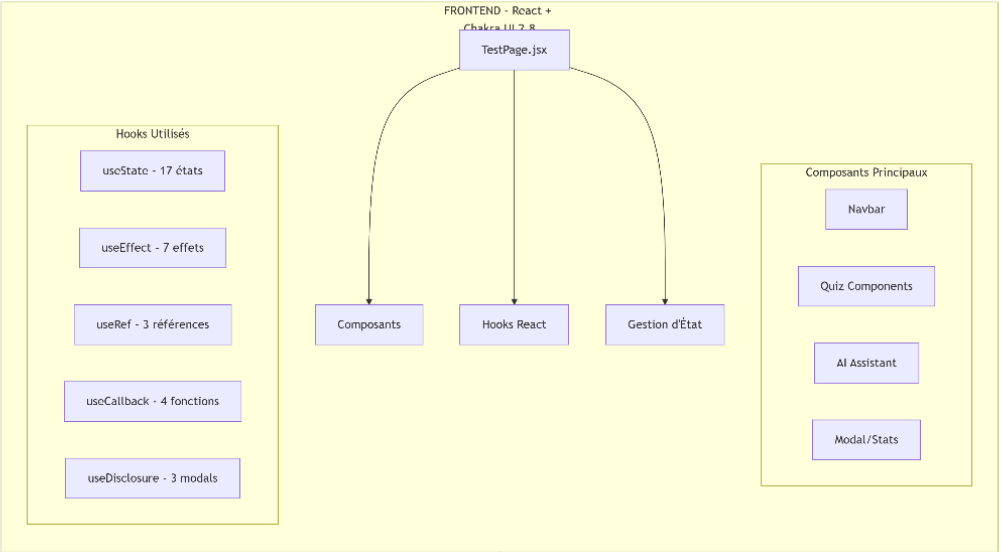
 **Animations :** Confetti, sons, effets visuels

## 2.3. Tableau de Bord Analytique 📊



- Score par catégorie (ligne de tendance)
- Taux de réussite global
- Progression XP (graphique aire)
- Points forts/faibles identifiés
- Recommandations personnalisées

### 3. ARCHITECTURE TECHNIQUE



### 3.1 Hooks React Utilisés

Hook	Problème Résolu	Solution Apportée	Exemple Concret	Impact
<code>useState</code> (17 utilisations)	"Comment stocker et mettre à jour des données qui changent ?" ❌ Variables normales ne rafraîchissent pas l'UI	<b>Stockage d'état réactif</b> ✓ Valeur + setter fonction ✓ Re-render automatique quand la valeur change	<pre>const [score, setScore] = useState(0); setScore(score + 10); // UI se met à jour</pre>	★★★★★ Fondamental
<code>useEffect</code> (7 utilisations)	"Comment exécuter du code quand quelque chose change ?" ❌ Code exécuté à chaque rendu = inefficace	<b>Exécution conditionnelle</b> ✓ S'exécute après le rendu ✓ Seulement quand dépendances changent ✓ Cleanup intégré	<pre>useEffect(() =&gt; {&lt;br&gt;  fetchQuestions(); &lt;br&gt;}, [langId]); // Seulement si langId change</pre>	★★★★★ Essentiel
<code>useRef</code> (3 utilisations)	"Comment garder une valeur entre les rendus sans déclencher de re-render ?" ❌ Variables normales sont réinitialisées	<b>Référence persistante</b> ✓ Survit aux re-renders ✓ Modification directe ( <code>.current</code> ) ✓ Pas de re-render automatique	<pre>const timerRef = useRef(null); timerRef.current = setInterval(...); clearInterval(timerRef.current);</pre>	★★★★★ Optimisation
<code>useCallback</code> (4 utilisations)	"Pourquoi mes composants enfants se re-rendent inutilement ?" ❌ Fonctions recréées à chaque rendu = nouvelles props	<b>Mémoïsation de fonctions</b> ✓ Même fonction tant que dépendances inchangées ✓ Évite re-renders enfants inutiles	<pre>const handleAnswer = useCallback(&lt;br&gt;  (answer) =&gt; { /* logique */ },&lt;br&gt;  [questionId] // Recrée si questionId change&lt;br&gt;);</pre>	★★★★★ Performance
<code>useDisclosure</code> (Chakra UI)	"Gérer l'ouverture/fermeture des modals est répétitif et verbeux" ❌ État + handlers manuels à réécrire	<b>État UI prêt à l'emploi</b> ✓ <code>isOpen</code> , <code>onOpen</code> , <code>onClose</code> inclus ✓ Gestion d'état simplifiée ✓ Accessibilité intégrée	<pre>const { isOpen, onOpen, onClose } =   useDisclosure(); &lt;Modal isOpen={isOpen} onClose={onClose}&gt;</pre>	★★★ Productivité
<code>useColorModeValue</code> (Chakra UI)	"Comment adapter l'interface au mode sombre/clair simplement ?" ❌ Conditions ternaires complexes partout	<b>Valeurs conditionnelles par thème</b> ✓ Mode clair → première valeur ✓ Mode sombre → deuxième valeur ✓ Réactif au changement de thème	<pre>const bg = useColorModeValue("white",   "gray.800"); // white en clair, gray.800 en sombre</pre>	★★★ UX/UI

### 4. Intégration du Modèle AI Qwen2.5:0.5b

```
-MERN> ollama list
NAME                                ID                                SIZE    MODIFIED
qwen2.5:0.5b-instruct-q4_K_M      a8b0c5157701                    397 MB  2 days ago
```

#### 4.1 Spécifications du Modèle

- Taille : 0.5 milliard de paramètres
- Quantisation : Q4\_K\_M (4-bit)
- Type : Instruct (optimisé pour instructions)
- Mémoire requise : ~300 MB
- Contexte : 32K tokens
- Langues supportées : Multilingue
- Licence : Apache 2.0

#### 4.2 Configuration Ollama

# Installation du modèle  
ollama pull qwen2.5:0.5b-instruct-q4\_K\_M

# Démarrage du serveur Ollama  
ollama serve

```
const OLLAMA_URL = 'http://localhost:11434';  
const MODEL = 'qwen2.5:0.5b-instruct-q4_K_M'; |
```

## 5. Considérations de Sécurité

- Chiffrement : bcrypt pour les mots de passe
- Tokens JWT : expiration 24h, refresh tokens
- CORS : restrictions par domaine
- Helmet : sécurisation headers HTTP
- Rate Limiting : prévention des attaques brute force
- Validation : sanitization des entrées utilisateur
- **Conclusion**

Cette architecture propose une solution complète pour une application d'apprentissage de langues avec :

1. Frontend moderne avec React et Chakra UI
2. Backend robuste avec Express.js et MongoDB
3. Intelligence Artificielle intégrée via Ollama et Qwen2.5
4. Système de gamification complet (XP, badges, leaderboard)
5. Sécurité renforcée à chaque niveau
6. Évolutivité garantie par une architecture modulaire

Le modèle Qwen2.5:0.5b-instruct-q4\_K\_M offre un bon équilibre entre performance et ressources, fonctionnant sur la plupart des machines modernes avec seulement ~300MB de RAM requis.