

Plan de Présentation - État de l'art Frontend R3MOB

1. Contexte

Ce que je dois dire :

"Cette semaine, j'ai réalisé un audit complet du code existant de R3MOB pour évaluer l'état actuel et identifier les axes d'amélioration prioritaires. Mon analyse s'est concentrée sur trois aspects : les performances, la qualité du code, et les nouveaux besoins pour la gestion des publications scientifiques."

Points clés à mentionner :

- Architecture actuelle : React.js + Express.js + MySQL depuis 2023
 - Base de code déjà solide mais nécessitant des optimisations
 - Focus sur la préparation de l'intégration des publications scientifiques
-

2. Problématiques Critiques Identifiées

A. Problèmes de Performance

Ce que je peux dire : "J'ai identifié plusieurs points qui impactent directement l'expérience utilisateur et les performances du site :"

- **Re-rendus excessifs :** "Les composants de filtrage se re-rendent trop souvent, ce qui ralentit l'interface"
- **Appels API non optimisés :** "J'ai trouvé des boucles infinies dans certains useEffect qui créent des appels API redondants"
- **Filtrage côté client :** "Le filtrage des données se fait entièrement côté client, ce qui devient lent avec de gros volumes"
- **Absence de mémorisation :** "Pas d'utilisation d'useMemo pour les calculs coûteux"

B. Qualité du Code

Ce que je peux dire : "Au niveau de la qualité du code, plusieurs pratiques peuvent être améliorées :"

- **Gestion d'erreurs :** "Actuellement, les erreurs sont gérées par des alertes basiques au lieu d'une interface utilisateur propre"
 - **Code redondant :** "Beaucoup de duplication dans les handlers d'événements"
 - **useEffect mal optimisés :** "Dépendances mal gérées dans les useEffect"
-

3. Solutions Techniques Recommandées

A. Choix Technologique : Maintenir React.js

Ce que je peux dire : "Après analyse, je recommande fortement de rester sur React.js car :"

- La base de code existante depuis 2023 représente un investissement important

- L'écosystème React est particulièrement adapté aux interfaces académiques complexes
- Excellente intégration avec les APIs scientifiques (Crossref, Semantic Scholar)

B. Optimisation avec React Query

Ce que je peux dire : "Pour résoudre les problèmes d'API, je propose d'intégrer React Query qui apporte :"

- Cache automatique des requêtes (réduction drastique des appels API)
- Synchronisation intelligente avec les APIs externes
- Gestion avancée des états de chargement et d'erreur
- Retry automatique pour la robustesse

4. Améliorations Prioritaires

A. Améliorations Techniques Immédiates

Ce que je peux dire : "J'ai identifié pas mal de pistes d'améliorations qui peuvent être implémentés au début :"

1. **Configuration** : "Utiliser un fichier .env pour les ports et configurations"
2. **Affichage** : "Corriger les problèmes CSS (images déformées, alignement header)"
3. **Documentation** : "Créer un setup Docker pour standardiser l'environnement de développement"

B. Fonctionnalités Publications Scientifiques

Ce qu'on peut dire : "Pour les nouvelles fonctionnalités publications, j'ai défini une approche progressive dont il faudra discuter par la suite avec celui qui s'occupe de l'import des publications"

1. **Import de fichiers** : "Parser BibTeX, XML, JSON côté client avec des librairies spécialisées"
2. **Recherche enrichie** : "Interface de saisie DOI/mots-clés avec interrogation Crossref"
3. **Affichage intelligent** : "Listes filtrables et triables suivant le modèle projets/chercheurs existant"
4. **Gestion d'erreurs** : "Messages clairs pour DOI invalides ou métadonnées manquantes" (cote Front)

5. Gestion du référencement et de la portée du site

Ce que vous devez dire :

"Pour améliorer le référencement et la portée du site, je propose :"

SEO et Visibilité :

- Optimisation des balises meta et mots-clés
- Génération automatique de sitemap XML (avec les toutes les routes du site)
- Structure URL optimisée pour le référencement

Langues :

- Version anglaise du site (comme R3IA et R3 Tesna)
- Sélecteur de langue FR/EN visible
- Traduction complète et cohérente du contenu

6. Sécurité et Maintenance

Ce que je peux dire :

"Pour la robustesse et la sécurité du système :"

Sécurité moderne :

- Implémentation de refresh tokens
- Intégration OAuth 2.0 via ORCID (standard recherche)
- RBAC (Role-Based Access Control)
- Protection XSS renforcée

Maintenance automatisée :

- npm audit pour les vulnérabilités
 - Dependabot pour les mises à jour automatiques
-

Phase 1 :

- Correction des problèmes de performance critiques
- Intégration React Query
- Améliorations CSS et configuration

Phase 2 :

- Développement du module publications scientifiques
- Implémentation de la partie Portée du site avec les langues

Phase 3 :

- Optimisations SEO avancées
- Renforcement sécurité