

NOTE DE CADRAGE - PROJET WRAPPER AI (Version Complète)

1. DÉFINITION ET OBJECTIFS

1.1 Contexte

Actuellement, les ingénieurs commerciaux perdent environ **30% de leur temps** à rechercher des informations techniques dispersées dans d'anciens dossiers archivés ou des contenus multimédias non indexés. Cette tâche manuelle est source d'erreurs et de frustration. L'entreprise **Tech Build Solutions** souhaite optimiser ce processus pour permettre aux équipes de se concentrer sur la valeur ajoutée.

1.2 Objectifs du projet

L'objectif est de développer un "Wrapper AI" polyvalent capable d'assister les équipes avant-vente et marketing via l'analyse documentaire et multimédia.

- **Objectif 1 : Accélération de la Phase d'Avant-Vente (Productivité)**
 - **But** : Réduire drastiquement le temps de recherche et de rédaction.
 - **Moyen** : Automatisation RAG (Retrieval-Augmented Generation).
 - **Mesure & Calcul Cible** :
 - *Formule* : $(\text{Temps manuel} - \text{Temps AI}) \times \text{Nombre de dossiers}$
 - *Projection* : Pour 200 appels d'offres/an, passer de 4h (manuel) à 30 min (AI) par recherche.
 - *Gain estimé* : $(4h - 0.5h) \times 200 = 700$ heures économisées/an (soit ~0.5 ETP).
- **Objectif 2 : Valorisation du Capital Cognitif Multimédia (Knowledge Management)**
 - **But** : Rendre exploitables les connaissances techniques "dormantes" (vidéos).
 - **Moyen** : Module "Repurpose" (Transcription + Synthèse).
 - **Mesure & Calcul Cible** :
 - *Indicateur* : Volume de "Dark Data" converti en texte indexable.
 - *Projection* : Traitement de 50 webinaires de 1h existants.
 - *Résultat* : $50h \text{ vidéo} \times \sim 10\,000 \text{ mots/h} = 500\,000 \text{ mots}$ de connaissances techniques ajoutés à la base.
- **Objectif 3 : Standardisation et Fiabilité de l'Information (Qualité)**
 - **But** : Éliminer les "hallucinations" et l'hétérogénéité.
 - **Moyen** : Citations obligatoires et Templates.
 - **Mesure & Calcul Cible** :
 - *Formule* : $1 - (\text{Nombre de réponses erronées} / \text{Total réponses})$
 - *Cible* : Atteindre un **Taux de Confiance > 98%** (Tolérance < 2 erreurs sur 100 requêtes grâce au sourcing).
- **Objectif 4 : Continuité de Service et Robustesse (Résilience Technique)**
 - **But** : Garantir la disponibilité malgré les pannes d'API externes.
 - **Moyen** : Architecture de Fallbacks (Gemini > Groq ; YouTube > Whisper).
 - **Mesure & Calcul Cible** :

- **Formule :** $(\text{Requêtes réussies via Fallback} / \text{Total pannes API principales}) * 100$
- **Cible : 99% de résilience.** Si Gemini échoue 100 fois, Groq doit prendre le relais avec succès au moins 99 fois.

1.3 Livrables attendus

- Note de cadrage validée.
- Solution technique fonctionnelle (Application Web Next.js).
- Documentation technique et utilisateur.

2. PÉRIMÈTRE (SCOPE)

2.1 Inclus dans le périmètre

Le projet s'étend au-delà de la simple assistance rédactionnelle :

- **Interface d'administration & Workspace :** Gestion des projets, upload de documents de référence et organisation par espaces de travail.
- **Moteur RAG (Chatbot) :** Interrogation en langage naturel de la base documentaire (CCTP, Docs techniques) avec citation des sources.
- **Module "Repurpose" :** Transformation intelligente de contenus vidéo (YouTube) en articles de blog, posts LinkedIn ou résumés techniques.
- **Export :** Génération de textes formatés (Markdown/HTML) prêts à l'emploi.

2.2 Exclus du périmètre

- Le chiffrage financier automatique complexe.
- L'intégration native dans des ERP lourds (SAP, Salesforce) – *Export manuel privilégié.*
- Application mobile native.

3. CONTRAINTES

3.1 Contraintes de calendrier

Le projet doit respecter strictement le calendrier universitaire :

- **Fin du cadrage (IT245) :** 28/01/2026.
- **Lancement de la réalisation (PR219) :** 28/01/2026.
- **Livraison finale et soutenance :** 01/04/2026.

3.2 Contraintes techniques et budgétaires

- **Budget :** 0€ (Projet étudiant). Utilisation de versions gratuites ou de crédits API limités.
- **Confidentialité :** Les données ne doivent pas servir à l'entraînement public des modèles IA (sélection de fournisseurs API conformes).
- **Accessibilité :** Solution web accessible via navigateur moderne.

4. ACTEURS ET GOUVERNANCE

- **Chef de Projet : Rémi LAMAT**
 - *Rôle :* Pilotage, respect des délais, coordination.
- **Équipe de réalisation : Estelle ABOU TAYEH, Tom DIETSCH, Mathieu MAGNIN**
 - *Rôle :* Développement Back-end, Front-end, Base de données.
- **Tuteur / Client (Intervenant) : Mr. Patrick PUJOL**

- *Rôle* : Validation des jalons et recette finale.
- **Cible finale** : Ingénieurs d'affaires, Directeur communication, Directeurs Techniques.

4.1 MATRICE RACI

(R = Réalise, A = Approuve/Responsable, C = Consulté, I = Informé)

Tâches / Livrables Clés	Chef de Projet (Rémi)	Équipe Dev (Estelle, Tom, Mathieu)	Tuteur (M. Pujol)
Rédaction Note de Cadrage	R	C	A (Valide)
Architecture & Choix Techno	A	R	C
Développement (Core/RAG)	I	R	I
Tests & Qualité Code	A	R	I
Validation Jalons (Recette)	R (Présente)	C	A (Valide)
Soutenance Finale	R	R	A (Évalue)

5. MACRO-PLANNING

Phase	Période	Livrables / Activités Clés
Phase 1 : Cadrage	13/01/2026 - 28/01/2026	Définition du besoin, étude de faisabilité, validation de la Note de Cadrage.
Phase 2 : Conception	28/01/2026 - 11/02/2026	Architecture technique, maquettes (Wireframes), choix des bibliothèques.
Phase 3 : Développement Core	11/02/2026 - 11/03/2026	Mise en place de la base vectorielle, ingestion des PDF, connexion API Gemini/Groq.
Phase 4 : Interface & Finalisation	11/03/2026 - 25/03/2026	Développement Front-end, intégration avec le Back-end, tests utilisateurs.
Phase 5 : Livraison	25/03/2026 - 01/04/2026	Recette finale, correction des bugs, préparation de la soutenance.

6. RESSOURCES (Mise à jour Technique)

L'architecture technique a été révisée pour correspondre à la réalité du développement et aux besoins de performance :

- **Ressources Humaines** : Équipe étudiante (Dev Fullstack & IA).
- **Stack Technique** :
 - **Frontend & Backend** : **Next.js 20** (App Router), React 19, TypeScript. Utilisation de TailwindCSS pour l'UI.
 - **Base de données & Auth** : **Supabase** (PostgreSQL avec pgvector pour la recherche sémantique).
 - **IA & LLM (Architecture Hybride)** :
 - Modèle Principal : **Google Gemini 1.5** (Performance/Coût).
 - Modèle Secondaire : **Groq** (Llama 3 / Mixtral) pour la rapidité et le secours.
 - **Traitement Média** : `youtubei.js` , `@distube/ytdl-core` pour l'extraction vidéo/audio.

7. PRÉCISIONS TECHNIQUES : RÉSILIENCE ET FALLBACKS

Pour répondre aux exigences de fiabilité (Objectif 3), l'application intègre des mécanismes de **fallback** (secours) automatiques :

1. **Fallback de Génération de Texte (LLM)** :
 - Le système tente d'abord de générer la réponse avec **Gemini** (Google).
 - En cas d'erreur (timeout, quota, indisponibilité), il bascule automatiquement sur **Groq** (modèles open-source rapides) pour assurer une réponse.
2. **Fallback d'Extraction de Contenu (YouTube)** :
 - L'application tente de récupérer la transcription via l'API YouTube standard.
 - En cas d'échec, elle bascule sur une méthode de scraping (`ytdl`).
 - Dernier recours : Utilisation de **Whisper** pour générer le texte à partir de la piste audio brute si aucune transcription n'est disponible.

8. COMMUNICATION

- **Interne (Équipe) :** Communication continue via Whatsapp. Point de synchronisation au début de chaque séance de TP.
- **Externe (Tuteur) :**
 - Validation des jalons aux dates clés (voir planning).
 - Alertes immédiates en cas de blocage technique majeur.
- **Documentation :** Les documents techniques et fonctionnels sont stockés sur le dépôt Git et Google Drive partagé.

9. MATRICE DES RISQUES (DÉTAILLÉE)

Catégorie	Risque Identifié	Prob.	Impact	Stratégie d'Atténuation & Plan B
Technique	Instabilité / Quota APIs Tiers (Gemini, Groq, YouTube)	Fort	Bloquant	Architecture Résiliente : Mise en place du circuit de fallback (Gemini → Groq → OpenAI). Plan B : Cache agressif des réponses et utilisation de données mockées pour la démo si panne le jour J.
Technique	Blocage Scraping YouTube (Changement DOM/Bot detection)	Moyen	Élevé	Redondance : Utilisation mixte API Officielle + ytdl + puppeteer. Plan B : Upload manuel du fichier audio/vidéo par l'utilisateur si le lien échoue.
Organisation	Surcharge de travail équipe (Examens, autres projets)	Moyen	Retard	Méthode Agile : Sprints courts (1 semaine) avec priorisation MoSCoW. On coupe les fonctionnalités "Nice to have" (ex: Export LinkedIn) pour sauver le "Core".
Fonctionnel	Hallucinations du RAG (Réponses fausses)	Moyen	Crédibilité	Ancrage Contextuel : Prompt système strict ("Réponds uniquement selon le contexte"). Affichage obligatoire des sources citées pour vérification humaine.
Juridique	Confidentialité des données (CCTP Clients)	Faible	Critique	Sécurité : Aucune persistance des données clients sur les serveurs d'IA (Opt-out training). Utilisation de Supabase RLS pour cloisonner les accès par utilisateur.

10. INDICATEURS DE RÉUSSITE (KPI MÉTIER)

(Mesure la valeur du produit fini)

- **Gain de Productivité Réel** : $(\text{Temps moyen processus manuel} - \text{Temps moyen processus Wrapper}) / \text{Temps manuel}$
 - *Cible* : > 50% de gain (ex: passer de 4h à 2h pour une synthèse).
- **Taux de Succès Technique (Fiabilité)** : $(\text{Nombre de requêtes réussies} / \text{Nombre total de requêtes}) * 100$
 - *Cible* : > 95% (grâce aux mécanismes de retry/fallback).
- **Qualité de la Transcription** : Taux de mots correctement transcrits (Word Error Rate estimé) via comparaison Whisper vs Script YouTube.
 - *Cible* : Compréhension du contexte > 90%.

11. INDICATEURS DE PILOTAGE DE PROJET (KPI PROJET)

(Mesure la santé du déroulement du projet)

- **Écart de Délai (Schedule Variance)** : $\text{Jours consommés} - \text{Jours planifiés}$ pour chaque jalon.
 - *Alerte* : Si retard > 3 jours sur une phase critique (Dev Core).
- **Taux d'Avancement Fonctionnel** : $(\text{Fonctionnalités "Must Have" livrées} / \text{Total "Must Have"}) * 100$
 - *Objectif* : 100% des fonctionnalités critiques livrées pour la soutenance.
- **Densité de Bugs (Qualité Code)** : Nombre de bugs bloquants ou majeurs ouverts en fin de sprint.
 - *Cible* : 0 bug bloquant 48h avant la livraison finale.
- **Consommation Budget API** : Suivi du nombre de tokens utilisés vs Quota gratuit disponible.
 - *Action* : Rotation des clés API si seuil de 80% atteint.

12. BIBLIOGRAPHIE ET LIENS KPI

- **Méthodologie de suivi** : [KPI pour évaluer un projet \(Wimi\)](#).
- **Gestion de projet** : [Quels sont vos KPI de suivi ?](#)
- **Cadrage** : [Réaliser une note de cadrage \(Manager-go\)](#).