



📄 GUIDE D'EXÉCUTION IA – STABILISATION GLOBALE & FINALISATION PRODUIT

Version : 1.1

Statut : Document d'exécution stricte

Destinataire : Copilot (mode développeur)

Projet : JETC_IMMO

Environnement : Supabase + Frontend (Next / HTML)

Connexion obligatoire : .env.local → DATA_URL, clés Supabase

0. RÈGLES ABSOLUES D'EXÉCUTION (NON NÉGOCIABLES)

1. Tu travailles **uniquement** sur l'existant.
 2. Tu te connectes à Supabase **via .env.local** (aucune clé en dur).
 3. Tu **ne supposes jamais** :
 - tu vérifies
 - tu montres la preuve
 - tu corriges
 4. Tu n'avances **pas à l'étape suivante** tant que la précédente n'est pas validée.
 5. Chaque correction doit produire :
 - un log
 - un résultat observable
 - une justification claire
 6. Aucun “fix rapide” sans comprendre **la cause racine**.
-

1. OBJECTIFS PRODUIT FINAUX (VISION CIBLE)



Fonctionnel global



Technicien

- Voir ses missions assignées
- Démarrer une mission
- Terminer une mission

- Ajouter des notes / rapport
- Signaler un incident
- Ajouter des photos
- Voir toutes les informations utiles (ticket, logement, locataire, accès)

Entreprise

- Voir les missions de ses techniciens
- Voir les missions terminées
- Créer une facture à partir d'une mission terminée
- Envoyer la facture à la régie via la plateforme

Régie

- Voir les interventions liées à ses biens
- Recevoir les factures des entreprises
- Avoir une vision claire des coûts

Admin JETC

- Voir toutes les régies / entreprises / techniciens / locataires
- Voir les interventions par régie
- Éditer les factures mensuelles détaillées
- Appliquer automatiquement la commission JETC (2 %)
- Voir les connexions en temps réel
- Avoir une vue globale stable et fiable

2. ÉTAPE 1 – AUTHENTIFICATION & STABILITÉ DES CONNEXIONS

2.1 Audit Auth Supabase

Tu dois vérifier **toutes** les vues :

- technicien
- entreprise
- régie
- admin

Pour chacune :

- comment la session est récupérée
- si supabase.auth.getSession() est utilisé correctement
- si plusieurs clients Supabase existent (interdit)
- si des routes API perdent la session

2.2 Bug critique actuel

POST /api/missions/start → 401 Unauthorized

Error: Non authentifié

Tu dois identifier :

- si le cookie/session est absent
- si la route API relit mal la session
- si un middleware bloque
- si le token expire ou n'est pas partagé entre pages

2.3 Résultat attendu

- Un utilisateur connecté reste connecté
- Plusieurs pages ouvertes → **une seule session**
- Aucune action légitime ne renvoie 401

3. ÉTAPE 2 – MODÈLE DE DONNÉES (SOURCE DE VÉRITÉ)

3.1 Relations officielles (à confirmer par SQL)

auth.users.id

→ profiles.id

→ techniciens.profile_id

→ missions.technicien_id

missions.ticket_id → tickets.id

tickets.logement_id → logements.id

logements.locataire_id → locataires.id

⚠️ Règle clé

- tickets.technicien_id peut rester NULL
- **La mission est la source de vérité pour l'exécution**

3.2 Vérifications SQL obligatoires

- missions sans technicien
- missions sans ticket
- tickets sans logement
- logements sans locataire
- clés étrangères manquantes

Aucune modification tant que l'audit n'est pas clair.

4. ÉTAPE 3 – WORKFLOW TECHNICIEN (CŒUR DU SYSTÈME)

4.1 États de mission autorisés

a_faire → en_cours → terminee

4.2 Actions autorisées

Action	Autorisé	Conditions
Voir mission	<input checked="" type="checkbox"/>	assigné
Démarrer mission	<input checked="" type="checkbox"/>	statut = a_faire
Ajouter notes	<input checked="" type="checkbox"/>	a_faire / en_cours
Ajouter photos	<input checked="" type="checkbox"/>	en_cours / terminee
Signaler incident	<input checked="" type="checkbox"/>	sauf annulée
Terminer mission	<input checked="" type="checkbox"/>	en_cours
Modifier ticket	<input checked="" type="checkbox"/>	jamais

4.3 Bug “Démarrer mission”

Tu dois :

1. Identifier la route exacte appelée

2. Logger payload + réponse
 3. Vérifier la policy RLS UPDATE sur missions
 4. Corriger **sans élargir les droits**
-

5. ÉTAPE 4 – RLS (ZÉRO COMPROMIS)

5.1 Technicien

- SELECT uniquement ses missions
- UPDATE uniquement ses missions
- INSERT signalements / photos liés à ses missions

5.2 Entreprise

- SELECT missions de son entreprise
- UPDATE factures après mission terminée

5.3 Admin

- SELECT global
- UPDATE factures / statistiques / régies

Chaque policy doit être :

- justifiée
 - lisible
 - testée
-

6. ÉTAPE 5 – FIN DE MISSION → FACTURATION

6.1 Flux attendu

1. Technicien termine la mission
2. Mission passe terminee
3. Entreprise voit mission facturable
4. Entreprise crée facture :
 - main d'œuvre
 - matériel

- TVA
5. Facture envoyée à la régie
 6. Admin :
 - consolide mensuellement
 - applique commission JETC 2 %

6.2 Audit existant

- Tables factures
 - factures_commissions_jtec
 - liens mission → facture
-

7. ÉTAPE 6 – GESTION DE LA LANGUE (OBLIGATOIRE)

7.1 Problème actuel

- **Seule la page d'accueil est traduite**
- Les dashboards, modales, boutons, statuts, erreurs ne le sont pas

7.2 Objectif

👉 **Tout le projet doit être multilingue**, toutes les vues, tous les rôles.

7.3 Source de vérité

- profiles.language (fr, en, etc.)
- Fallback si non défini

7.4 À vérifier et corriger

- Tous les textes UI (boutons, titres, statuts)
- Messages d'erreur
- Modales
- Emails (préparation)
- Libellés de statuts mission / facture

7.5 Règle

Aucun texte en dur dans les vues métier.

Tout doit passer par le système de traduction.

8. ÉTAPE 7 – VUE ADMIN (JETC)

8.1 Dashboard requis

- Nombre en temps réel :
 - régies
 - entreprises
 - techniciens
 - locataires
 - propriétaires
- Interventions par régie
- Factures mensuelles détaillées
- Commission 2 %

8.2 Temps réel

- Vérifier Supabase Realtime
- Connexions actives par rôle
- Pas de polling inutile

9. ÉTAPE 8 – EMAILS (PRÉPARATION UNIQUEMENT)

- Vérifier :
 - templates existants
 - logique de génération de login
- **Ne pas activer l'envoi**
- Documenter ce qui est prêt et ce qui manque

10. LIVRABLES OBLIGATOIRES

Copilot doit produire :

1. RAPPORT_GLOBAL_AUDIT_JETC_IMMO.md
2. Liste des bugs (cause → correction)

3. Liste des fichiers modifiés
 4. Migrations SQL appliquées
 5. Tests effectués
 6. Liste des points restants (si existants)
-

11. RÈGLE FINALE

Si quelque chose n'est pas clair :

- tu t'arrêtes
- tu expliques
- tu proposes
- tu n'exécutes pas