THE VIBE CODEUR

LE GUIDE OFFERT

TON PREMIER ÉCRAN EN 30 MINUTES

40 prompts pour passer d'une idée floue à une application qui tourne vraiment — sans écrire une ligne de code. Chacun expliqué : quand l'utiliser, pourquoi il est écrit ainsi, et ce que tu dois obtenir.

CAMEL DAGBEGNON

The Vibe Codeur · Dagbegnon Agency · Bénin

masterclass.
dagbegnonagency.com

AVANT DE COMMENCER — DEUX PAGES À LIRE

Ce guide ne t'apprend pas à coder. Il t'apprend à **dire précisément ce que tu veux** à une IA, à reconnaître quand elle se trompe, et à avancer sans jamais rester bloqué plus de dix minutes.

COMMENT CHAQUE FICHE EST CONSTRUITE

Un prompt copié sans être compris ne sert qu'une fois. Chaque fiche te donne donc quatre choses :

BLOC	CE QU'IL TE DONNE
Quand t'en servir	Le moment exact du projet, et le niveau requis
Le prompt	À copier tel quel. Les mots EN MAJUSCULES sont à remplacer par les tiens
◆ Pourquoi il est écrit comme ça	La technique derrière. C'est ce bloc qui te rendra autonome
✓ Ce que tu dois obtenir	Comment savoir que ça a marché, sans deviner
! Si ça ne marche pas	Quoi faire exactement, plutôt que de recommencer au hasard

LES TROIS RÈGLES QUI CHANGENT TOUT

1 **Un prompt = une seule chose**
Demande un écran, pas une application. Un bouton, pas un écran entier. Quand tu demandes trop d'un coup, l'IA comble les vides en inventant — et tu passes la soirée à démêler ce qu'elle a supposé. Le petit morceau qui marche vaut mieux que le gros qui plante.

2 **Décris le résultat, pas la technique**
Ne dis pas « utilise un tableau relationnel indexé ». Dis « je veux voir la liste des commandes du jour, la plus récente en haut, et pouvoir cliquer dessus ». L'IA connaît la technique mieux que toi. Toi, tu connais le besoin mieux qu'elle. Chacun son domaine.

3 **Teste après chaque prompt, sans exception**
Ouvre, clique, casse. Si ça marche, tu passes au suivant. Si ça ne marche pas, tu corriges maintenant — pas dans dix prompts, quand tu ne sauras plus lequel a introduit le problème.

LE NIVEAU INDIQUÉ SUR CHAQUE FICHE

Débutant	Tu copies, tu colles, tu regardes le résultat. Aucun prérequis.
Intermédiaire	Tu dois comprendre ce que tu lis pour vérifier la réponse.
Exigeant	Une erreur ici coûte de l'argent ou des données. Prends ton temps.

LE PARCOURS COMPLET

Phase 1	Cadrer l'idée	prompts 01–06

Phase 2	Le premier écran	07–14
Phase 3	Les données	15–21
Phase 4	Les comptes utilisateurs	22–27
Phase 5	Encaisser en FCFA	28–32
Phase 6	Mettre en ligne	33–36
Phase 7	Quand ça casse	37–40

UNE RÈGLE D'HONNÊTETÉ

Ces prompts sont des points de départ éprouvés sur de vrais projets, pas des formules magiques. L'IA se trompera, et elle se trompera avec assurance. Ton travail est de vérifier — c'est pour ça que chaque fiche te dit précisément quoi vérifier, et que la phase 7 est entièrement consacrée au déblocage.

PHASE 1 — CADRER L'IDÉE

C'est l'étape que tout le monde saute, et c'est celle qui fait abandonner. Six prompts pour savoir exactement quoi construire *avant* d'ouvrir quoi que ce soit. Compte vingt minutes. Elles t'en feront gagner vingt heures.

01 Transformer une idée floue en une phrase

NIVEAU : DÉBUTANT | TOUT AU DÉBUT | ~5 MIN

Quand t'en servir. Tu as « une idée d'application » mais tu n'arrives pas à l'expliquer en dix secondes à quelqu'un qui ne connaît rien au sujet. Tant que cette phrase n'existe pas, chaque décision technique sera un coup de dés.

```
Tu es un consultant produit qui aide des débutants à clarifier leur idée.

Voici mon idée, en vrac et mal formulée :
DÉCRIS TON IDÉE AVEC TES MOTS, MÊME EN DÉSORDRE

Reformule-la en une seule phrase de cette forme exacte :
« Une application qui permet à [QUI PRÉCISÉMENT] de [FAIRE QUOI] pour [QUEL BÉNÉFICE
CONCRET] ».

Donne-moi trois versions qui diffèrent vraiment (pas trois synonymes).
Puis dis-moi laquelle est la plus précise et pourquoi, en te basant sur
ce critère : est-ce qu'un inconnu comprendrait à qui ça sert, sans explication ?
Si mon idée est trop vague pour que tu puisses répondre, dis-le et pose-moi
les questions qui te manquent.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

La forme imposée « [QUI] / [QUOI] / [BÉNÉFICE] » empêche l'IA de te rendre une phrase creuse du genre « une plateforme innovante ». La demande de trois versions *qui diffèrent vraiment* évite les variations cosmétiques. Et la dernière ligne autorise l'IA à te dire qu'elle manque d'information, au lieu d'inventer — c'est la ligne la plus utile du prompt.

✓ **CE QUE TU DOIS OBTENIR**

Trois phrases où tu peux nommer une personne réelle de ton entourage qui correspond au « QUI ». Si tu n'y arrives pour aucune des trois, ton idée est encore trop large.

! **SI ÇA NE MARCHE PAS**

Les phrases restent vagues ? C'est que ta description de départ l'était. Réponds aux questions que l'IA te pose et relance le même prompt en y ajoutant tes réponses.

02 Trouver le plus petit morceau livrable

NIVEAU : DÉBUTANT | APRÈS LE 01 | ~5 MIN

Quand t'en servir. Juste après avoir votre phrase. C'est le prompt le plus important du guide : un projet abandonné est presque toujours un projet trop gros pour le temps qu'on avait.

```
Voici mon projet : COLLE LA PHRASE RETENUE AU PROMPT 01

Découpe-le en fonctionnalités distinctes. Puis identifie LA SEULE qui,
livrée toute seule, rend déjà un service réel à quelqu'un — même si le reste
n'existe pas.

Applique ce test à chaque candidate : « si je ne livrais que ça, est-ce que
quelqu'un l'utiliserait quand même ? ». Écarte toutes celles qui échouent.

Pour celle que tu retiens, dis-moi :
— pourquoi c'est celle-là et pas une autre
— ce qu'on perd concrètement à commencer par une autre
— combien de temps un débutant met raisonnablement à la construire

Ne me propose pas de « version minimale » qui contiendrait encore
quatre fonctionnalités. J'en veux une.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Le « test » explicite donne à l'IA un critère mesurable au lieu de son intuition. Et la dernière phrase contre-attaque un travers connu : quand on demande un produit minimal, les modèles proposent presque toujours un ensemble de quatre ou cinq fonctions. Il faut le leur interdire nommément.

✓ **CE QUE TU DOIS OBTENIR**

Une seule fonctionnalité, descriptible en une phrase, avec une estimation de temps qui ne dépasse pas quelques jours. Si l'estimation parle de semaines, redécoupe.

! **SI ÇA NE MARCHE PAS**

L'IA insiste pour en garder plusieurs ? Réponds simplement : « Choisis-en une seule. Si tu dois te tromper, trompe-toi du côté du plus petit. »

03 Lister les écrans nécessaires

NIVEAU : DÉBUTANT | APRÈS LE 02 | ~3 MIN

Quand t'en servir. Avant de dessiner ou de générer quoi que ce soit. Connaître le nombre d'écrans, c'est connaître la taille réelle du travail.

```
Pour cette seule fonctionnalité : COLLE LA RÉPONSE DU PROMPT 02

Liste le nombre MINIMAL d'écrans nécessaires. Pour chacun :
— son nom
— ce qu'on y voit
— ce qu'on peut y faire
— vers quel autre écran il mène

Contrainte : si tu proposes plus de quatre écrans, recommence en fusionnant
ou en supprimant, et explique-moi ce que tu as sacrifié.
Présente le résultat sous forme de tableau.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Vers quel autre écran il mène » force l'IA à vérifier que ton parcours tient debout : un écran sans sortie est un écran oublié. La contrainte des quatre écrans n'est pas arbitraire — au-delà, un débutant perd le fil avant d'avoir fini.

✓ **CE QUE TU DOIS OBTENIR**

Un tableau de deux à quatre lignes, où tu peux suivre du doigt le chemin d'un écran à l'autre sans impasse.

! **SI ÇA NE MARCHE PAS**

Plus de quatre écrans malgré la consigne ? C'est un signal : reviens au prompt 02, ton morceau est encore trop gros.

04 Décrire celui qui va s'en servir

NIVEAU : DÉBUTANT | EN PARALLÈLE DU 03 | ~4 MIN

Quand t'en servir. Avant de choisir la moindre couleur ou le moindre mot d'interface. Sans ça, tu construiras pour toi — et tu es la personne la moins représentative de tes utilisateurs.

```
L'utilisateur de mon application est : DÉCRIS-LE (métier, âge approximatif,
à quel point il est à l'aise avec un téléphone, dans quel contexte il s'en sert)

Sur cette base, dis-moi :
1. les trois mots ou concepts qu'il ne comprendra pas si je les emploie tels quels
2. le moment précis du parcours où il risque d'abandonner, et pourquoi
3. sur quel appareil et dans quelles conditions il l'utilisera le plus souvent
   (connexion lente ? debout ? en pleine rue ? une main occupée ?)
4. une chose que je crois évidente et qui ne le sera pas pour lui

Sois concret et sévère. Ne me rassure pas.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Les questions numérotées obligent à une réponse structurée plutôt qu'à un portrait littéraire inutile. « Ne me rassure pas » contre la tendance des modèles à être agréables : tu veux des objections, pas des compliments. Le point 3 est celui qu'on oublie toujours, et c'est lui qui décide de la taille des boutons.

✓ **CE QUE TU DOIS OBTENIR**

Au moins une réponse qui te surprend ou te dérange. Si tout te paraît évident, c'est que tu as décrit quelqu'un qui te ressemble trop.

! **SI ÇA NE MARCHE PAS**

Réponses génériques ? Ta description de départ manquait de contexte. Ajoute un exemple réel : « c'est ma tante qui tient une boutique et qui note tout sur un cahier ».

05 Écrire le parcours en six étapes

NIVEAU : DÉBUTANT | APRÈS LE 03 ET LE 04 | ~4 MIN

Quand t'en servir. Dernière vérification avant de construire. Le parcours est ce que l'utilisateur vivra ; les écrans n'en sont que le décor.

```
Écris le parcours complet de mon utilisateur, de son arrivée jusqu'au moment
où il a obtenu ce qu'il voulait.

Contraintes :
— six étapes maximum, une phrase chacune
— à chaque étape, précise sur quel écran il se trouve
— formule chaque étape du point de vue de l'utilisateur, pas du système
  (« il voit », « il tape », « il reçoit » — jamais « le système enregistre »)

Puis signale-moi :
— toute étape où il pourrait se perdre ou hésiter
— toute étape où il doit faire confiance sans preuve (paiement, envoi de données)
— la seule étape que je pourrais supprimer sans casser le parcours
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Imposer le point de vue de l'utilisateur (« il voit ») empêche de décrire la mécanique interne, qui ne t'apprend rien sur l'expérience. La question des étapes « où il doit faire confiance » est la plus rentable du guide : c'est exactement là que les gens abandonnent.

✓ **CE QUE TU DOIS OBTENIR**

Six phrases que tu pourrais lire à voix haute à quelqu'un, qui hocherait la tête. Plus une étape identifiée comme supprimable.

! **SI ÇA NE MARCHE PAS**

Le parcours dépasse six étapes ? Demande : « Fusionne les étapes 3 et 4, et dis-moi ce que ça coûte en clarté. »

06 Éliminer tout ce qui n'est pas la version 1

NIVEAU : DÉBUTANT | FIN DE PHASE 1 | ~6 MIN

Quand t'en servir. Le dernier filtre avant d'ouvrir un éditeur. Vide ta tête de toutes les idées qui t'empêchent de commencer.

```
Voici tout ce que j'imagine, un jour, pour mon application :
LISTE TOUT, MÊME LES IDÉES QUE TU TROUVES SECONDAIRES

Range chaque élément dans exactement une de ces trois colonnes :
— INDISPENSABLE : sans ça, l'application ne rend aucun service
— PLUS TARD : utile, mais l'application marche sans
— JAMAIS : ça n'apporte rien de mesurable

Règles :
— sois sévère : par défaut, un élément va dans PLUS TARD
— justifie chaque INDISPENSABLE en une phrase
— si la colonne INDISPENSABLE dépasse cinq éléments, recommence :
  tu as été trop généreux

Termine par la liste ordonnée de ce que je dois construire en premier.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Par défaut, un élément va dans PLUS TARD » inverse la charge de la preuve : c'est à une fonctionnalité de prouver qu'elle est indispensable, pas l'inverse. Le plafond de cinq éléments et l'ordre de construction final transforment une réflexion en plan d'action.

✓ **CE QUE TU DOIS OBTENIR**

Une colonne INDISPENSABLE de cinq lignes maximum, et une liste ordonnée par laquelle commencer dès aujourd'hui.

! **SI ÇA NE MARCHE PAS**

Tu n'arrives pas à trancher ? Demande : « Si je n'avais qu'un week-end, que garderais-tu ? » La contrainte de temps force la décision.

PHASE 2 — LE PREMIER ÉCRAN

> Objectif de la phase : quelque chose qui s'affiche et que tu peux montrer à quelqu'un. C'est ici que le projet cesse d'être une idée. Ne cherche pas la perfection — cherche l'écran qui existe.

07 Créer la structure du projet

NIVEAU : DÉBUTANT | PREMIER PROMPT TECHNIQUE | ~10 MIN

Quand t'en servir. Une fois ton périmètre figé au prompt 06. C'est le moment où tu ouvres un outil pour la première fois.

```
Crée la structure d'un projet web pour cette application :
COLLE LA PHRASE DU PROMPT 01 ET LA FONCTIONNALITÉ DU PROMPT 02

Contraintes importantes :
— je suis débutant complet, je n'ai jamais publié de site
— choisis les technologies les plus SIMPLES possibles, pas les plus modernes
— aucune configuration compliquée, aucun outil à installer si on peut l'éviter
— le projet doit pouvoir tourner en ouvrant un fichier ou avec une seule commande

Donne-moi :
1. la liste de tous les fichiers, avec en une ligne à quoi sert chacun
2. le contenu complet de chaque fichier
3. les étapes exactes, numérotées, pour le voir s'afficher dans mon navigateur
4. ce que je dois voir à l'écran si tout s'est bien passé

Si tu hésites entre deux approches, choisis la plus simple à dépanner
pour quelqu'un qui ne sait pas lire du code.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Les plus simples possibles, pas les plus modernes » est essentiel : sans cette contrainte, les modèles proposent des outils à la mode qui exigent une demi-journée d'installation. Le point 4 — « ce que je dois voir si ça a marché » — te donne un critère de réussite au lieu de te laisser deviner.

✓ **CE QUE TU DOIS OBTENIR**

Une page qui s'ouvre dans ton navigateur, même vide ou moche. Si tu vois quelque chose, la phase est réussie.

! **SI ÇA NE MARCHE PAS**

Une étape échoue ? Ne passe pas à la suivante. Copie le message exact et va au prompt 37, qui est fait pour ça.

08 Le premier écran qui s'affiche

NIVEAU : DÉBUTANT | APRÈS LE 07 | ~10 MIN

Quand t'en servir. Dès que la structure tourne. On construit l'apparence avant la mécanique : c'est plus motivant, et ça révèle les erreurs de conception plus tôt.

```
Crée l'écran « NOM DE L'ÉCRAN », décrit ainsi :
COLLE SA LIGNE DU TABLEAU DU PROMPT 03

Contraintes :
— utilise des données INVENTÉES écrites en dur dans le fichier, pas de base
  de données pour l'instant
— mets suffisamment de fausses données pour que je voie à quoi ça ressemble
  quand c'est rempli (au moins cinq lignes, avec des cas variés :
  un nom très long, un champ vide, une valeur à zéro)
— code lisible et commenté en français

Je veux uniquement voir l'apparence. On branchera le vrai plus tard.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Demander des cas *variés* — nom très long, champ vide, zéro — est une technique de professionnel. Avec trois fausses lignes bien rangées, tout paraît beau ; avec un nom de quarante caractères, tu découvres tout de suite que ta mise en page casse.

✓ **CE QUE TU DOIS OBTENIR**

Un écran rempli qui tient debout même avec le nom très long et la ligne vide. Si ça déborde, c'est une bonne nouvelle : tu viens de trouver un bug gratuitement.

! **SI ÇA NE MARCHE PAS**

Rien ne s'affiche ? Vérifie d'abord que tu as bien enregistré le fichier et rechargé la page. C'est la cause dans un cas sur deux.

09 Corriger la mise en page

NIVEAU : DÉBUTANT | DÈS QUE ÇA S'AFFICHE | ~5 MIN

Quand t'en servir. Quand ça fonctionne mais que c'est mal rangé. Un prompt à relancer autant de fois que nécessaire, un défaut à la fois.

```
Voici mon écran actuel :
COLLE LE CODE COMPLET

Ce qui ne va pas, précisément :
DÉCRIS CE QUE TU VOIS, PUIS CE QUE TU VOUDRAIS À LA PLACE

Contraintes :
— corrige UNIQUEMENT la mise en page
— ne change ni le fonctionnement, ni les textes, ni la structure des données
— explique-moi en une phrase ce que tu as modifié et pourquoi
— rends-moi le fichier complet, pas seulement l'extrait modifié
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Uniquement la mise en page » évite qu'un modèle serviable en profite pour « améliorer » le reste et casse ce qui marchait. Demander le fichier complet évite l'erreur classique du débutant : coller un extrait au mauvais endroit.

✓ **CE QUE TU DOIS OBTENIR**

Le défaut corrigé, et rien d'autre de changé. Recharge et vérifie que ce qui marchait marche encore.

! **SI ÇA NE MARCHE PAS**

Autre chose a cassé ? Dis-le franchement : « Tu as aussi modifié X, remets-le comme avant. »

10 Le rendre utilisable sur téléphone

NIVEAU : DÉBUTANT | TÔT, JAMAIS À LA FIN | ~8 MIN

Quand t'en servir. Dès le premier écran terminé. En Afrique francophone, la quasi-totalité de tes utilisateurs seront sur téléphone. Repousser cette étape, c'est tout refaire plus tard.

```
Rends cet écran réellement utilisable sur un téléphone de 360 pixels de large :
COLLE LE CODE

Exigences non négociables :
— aucun défilement horizontal, jamais
— texte à 16 pixels minimum dans les champs de saisie
  (en dessous, les iPhone zooment automatiquement et c'est très désagréable)
— zones cliquables d'au moins 44 pixels de haut
— rien de coupé en bas de l'écran, y compris sur les téléphones à encoche
— les boutons importants atteignables avec le pouce, donc plutôt en bas

Ensuite, liste-moi ce que je dois vérifier moi-même sur mon propre téléphone
pour confirmer que c'est bon.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Les chiffres précis (360 px, 16 px, 44 px) sont des seuils réels, pas des préférences : en dessous de 16 px, iOS zoome tout seul sur les champs ; en dessous de 44 px, le doigt rate la cible. Donner les seuils évite les « j'ai rendu ça responsive » qui ne veulent rien dire.

✓ **CE QUE TU DOIS OBTENIR**

Sur ton téléphone : aucun glissement latéral, aucun zoom involontaire en tapant dans un champ, et tous les boutons atteignables d'une main.

! **SI ÇA NE MARCHE PAS**

Ça déborde encore ? Demande : « Trouve quel élément dépasse la largeur de l'écran et corrige uniquement celui-là. »

11 Appliquer ton identité visuelle

NIVEAU : DÉBUTANT | QUAND LA STRUCTURE EST BONNE | ~6 MIN

Quand t'en servir. Une fois la mise en page stable. Avant, tu perdrais ton temps à colorier quelque chose qui va encore bouger.

```
Applique cette identité visuelle à toute l'application :
— couleur principale : TA COULEUR (code hexadécimal si tu l'as)
— fond : CLAIR OU SOMBRE
— ambiance recherchée : SÉRIEUX / CHALEUREUX / MODERNE / SOBRE

Exigence technique importante :
définis toutes les couleurs à UN SEUL endroit, sous forme de variables
réutilisées partout, pour que je puisse changer toute l'apparence
en modifiant quelques lignes.

Vérifie aussi que le texte reste lisible sur chaque fond
(contraste suffisant), et signale-moi toute combinaison douteuse.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Les variables centralisées sont ce qui sépare un projet qu'on peut faire évoluer d'un projet où changer une couleur demande de fouiller trente fichiers. La vérification du contraste évite le piège du gris clair sur blanc, illisible dehors en plein soleil — un vrai problème ici.

✓ **CE QUE TU DOIS OBTENIR**

Un bloc de couleurs identifiable en haut du fichier. Change-en une : toute la page doit suivre.

! **SI ÇA NE MARCHE PAS**

Les couleurs sont éparpillées ? Demande : « Regroupe toutes les couleurs en variables au même endroit, sans changer le rendu. »

12 Ajouter un formulaire

NIVEAU : DÉBUTANT | DÈS QU'IL Y A UNE SAISIE | ~8 MIN

Quand t'en servir. Au moment où l'utilisateur doit te donner quelque chose. C'est le point le plus fragile de toute application : chaque champ en trop fait perdre des gens.

```
Ajoute un formulaire qui demande exactement ces informations :
LISTE LES CHAMPS, ET POUR CHACUN S'IL EST OBLIGATOIRE OU FACULTATIF

Exigences :
— les champs facultatifs sont visiblement marqués « facultatif »
— le bouton d'envoi reste désactivé tant qu'un champ obligatoire est vide
— pendant l'envoi : le bouton affiche un état d'attente et devient incliquable
— après l'envoi : un message clair de réussite ou d'échec
— en cas d'échec, ce que l'utilisateur a saisi n'est JAMAIS effacé

Avant d'écrire le code, dis-moi si l'un de ces champs te paraît inutile
pour la première version, et pourquoi.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Ce qui a été saisi n'est jamais effacé » est la règle que les débutants découvrent trop tard : perdre la saisie d'un utilisateur après une erreur réseau, c'est perdre l'utilisateur. La dernière question retourne l'IA contre ton propre formulaire — elle trouve souvent un champ dont tu n'as pas besoin.

✓ **CE QUE TU DOIS OBTENIR**

Un formulaire où le bouton est gris tant que tu n'as pas rempli l'essentiel, et où couper ta connexion en plein envoi ne te fait pas tout retaper.

! **SI ÇA NE MARCHE PAS**

Le bouton reste actif ? Demande : « Désactive le bouton tant que les champs obligatoires ne sont pas valides, et montre-moi la ligne qui le fait. »

13 Vérifier ce qui est saisi

NIVEAU : INTERMÉDIAIRE | JUSTE APRÈS LE 12 | ~8 MIN

Quand t'en servir. Immédiatement après avoir créé le formulaire. Jamais plus tard : une fois en ligne, chaque saisie fautive devient une ligne à corriger à la main.

```
Ajoute la vérification des saisies de ce formulaire :
COLLE LE CODE

Règles précises :
— un e-mail doit ressembler à un e-mail ; signale poliment les fautes de frappe
  courantes dans les domaines (gmial.com, gmzil.com, hotmial.com…)
— un numéro de téléphone doit être accepté AVEC OU SANS indicatif international,
  avec ou sans espaces, points ou tirets
— n'exige JAMAIS un format national précis : un utilisateur d'un autre pays
  doit pouvoir s'inscrire
— les messages d'erreur sont en français, courts, et disent quoi corriger
  (pas « champ invalide », mais « il manque le @ dans ton adresse »)
— l'erreur s'affiche à côté du champ concerné, pas en haut de page

Vérifie aussi côté serveur, pas seulement dans le navigateur, et explique-moi
pourquoi les deux sont nécessaires.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Refuser quelqu'un parce que son numéro n'a pas le format attendu, c'est perdre un client pour rien — et c'est une erreur très fréquente dans les applications faites localement. La double vérification (navigateur et serveur) est le réflexe qui sépare une application jouet d'une application sérieuse : celle du navigateur se contourne en quelques secondes.

✓ **CE QUE TU DOIS OBTENIR**

Teste avec un numéro de ton pays, puis avec un numéro étranger précédé de son indicatif : les deux doivent passer. Teste une adresse en « @gmial.com » : elle doit être signalée sans être brutalement rejetée.

! **SI ÇA NE MARCHE PAS**

Un numéro valide est refusé ? Donne l'exemple exact à l'IA : « Ce numéro est valide et refusé : […]. Corrige la règle. »

14 Empêcher le double envoi

NIVEAU : EXIGEANT | AVANT TOUTE MISE EN LIGNE | ~10 MIN

Quand t'en servir. Avant que qui que ce soit d'autre que toi n'utilise l'application. Ce prompt t'évite le problème le plus embarrassant qui soit : deux commandes, deux débits, un client furieux.

```
Empêche qu'un double clic, un rechargement de page ou un retour en arrière
crée deux fois la même chose.

Voici mon formulaire et le code qui l'enregistre :
COLLE LE CODE

Mets en place les trois protections, et montre-moi où chacune agit :
1. côté navigateur : le bouton se désactive dès le premier clic
2. côté envoi : chaque soumission porte un identifiant unique, généré une
   seule fois par visite
3. côté serveur : si un identifiant déjà traité revient, renvoie le résultat
   précédent SANS rien enregistrer de nouveau

Explique-moi pourquoi la protection 1 seule ne suffit pas.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Les trois couches ne sont pas redondantes : le bouton désactivé ne protège pas d'un rechargement, et le rechargement ne protège pas d'une double requête réseau. Seul l'identifiant unique vérifié côté serveur garantit qu'une action ne compte qu'une fois. Demander à l'IA d'expliquer pourquoi la première ne suffit pas te permet de vérifier qu'elle a compris — et pas seulement obéi.

✓ **CE QUE TU DOIS OBTENIR**

Envoie le formulaire, puis recharge la page en confirmant le renvoi : une seule entrée doit exister. C'est le test, fais-le vraiment.

! **SI ÇA NE MARCHE PAS**

Deux entrées apparaissent ? L'identifiant est probablement régénéré à chaque envoi. Demande : « Génère l'identifiant une seule fois par visite et conserve-le. »

PHASE 3 — LES DONNÉES

> Jusqu'ici, tout était faux. À partir d'ici, ce que l'utilisateur saisit existe vraiment et doit survivre à la fermeture du navigateur. C'est aussi là que commencent les vraies responsabilités.

15 Décrire ce qu'il faut stocker

NIVEAU : INTERMÉDIAIRE · AVANT DE CRÉER QUOI QUE CE SOIT · ~8 MIN

Quand t'en servir. Avant de toucher à une base de données. Une structure mal pensée se paie pendant des mois ; bien pensée, elle ne se remarque jamais.

```
Mon application doit mémoriser durablement :
DÉCRIS AVEC TES MOTS CE QUI DOIT ÊTRE CONSERVÉ

Propose-moi la structure de données la plus simple qui tienne.
Pour chaque information : son nom, son type, si elle est obligatoire,
et un exemple de valeur réelle.

Puis, et c'est le plus important :
— signale-moi ce que j'ai oublié et qui me manquera dans trois mois
— dis-moi quelles informations je devrais horodater (date de création,
  date de modification) et pourquoi
— préviens-moi si une de mes informations est une donnée personnelle
  qui mérite d'être protégée

Ne propose pas de structure compliquée en prévision de besoins hypothétiques.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Ce qui me manquera dans trois mois » exploite ce que le modèle a vu dans des milliers de projets : il sait que tu regretteras de ne pas avoir daté tes enregistrements, même si toi tu n'y penses pas. La dernière ligne évite l'excès inverse, où l'on conçoit pour un million d'utilisateurs qu'on n'aura jamais.

✓ **CE QUE TU DOIS OBTENIR**

Un tableau que tu comprends ligne par ligne, avec au moins une information que tu avais oubliée — souvent une date.

! **SI ÇA NE MARCHE PAS**

La structure te paraît trop compliquée ? Demande : « Simplifie : que puis-je supprimer sans me bloquer plus tard ? »

16 Mettre en place le stockage

NIVEAU : INTERMÉDIAIRE | APRÈS VALIDATION DU 15 | ~20 MIN

Quand t'en servir. Une fois la structure validée, et pas avant. C'est l'étape la plus technique du guide : prends-la calmement.

```
Mets en place le stockage de cette structure :
COLLE LA RÉPONSE DU PROMPT 15

Contraintes :
— je suis débutant, je n'ai jamais installé de base de données
— choisis la solution GRATUITE la plus simple à mettre en route, quitte à
  ce qu'elle soit moins puissante
— pas d'installation lourde sur mon ordinateur si une solution en ligne existe

Donne-moi :
1. le nom de la solution retenue et pourquoi elle convient à mon cas
2. les étapes exactes de création du compte et de la base, numérotées,
   sans supposer que je connais le vocabulaire
3. où je dois coller les identifiants de connexion, et surtout
   comment éviter qu'ils se retrouvent publiés par accident
4. comment vérifier que la connexion fonctionne avant d'aller plus loin
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Le point 3 est le plus important de la fiche : un identifiant de base de données publié par mégarde, c'est ta base ouverte à tous. Demander explicitement comment l'éviter t'apporte la réponse avant l'accident, pas après. Le point 4 te donne un point de contrôle : ne jamais enchaîner sur du code applicatif tant que la connexion n'est pas prouvée.

✓ **CE QUE TU DOIS OBTENIR**

Une base créée, et un test simple qui confirme que ton application arrive à s'y connecter. Rien d'autre à ce stade.

! **SI ÇA NE MARCHE PAS**

La connexion échoue ? C'est presque toujours un identifiant mal copié ou une autorisation d'accès à activer. Copie le message exact et passe au prompt 37.

17 Enregistrer pour de vrai

NIVEAU : INTERMÉDIAIRE | APRÈS LE 16 | ~12 MIN

Quand t'en servir. Le moment où ton projet devient une vraie application. Remplace les fausses données par les vraies.

```
Branche mon formulaire sur le stockage, à la place des données inventées.

Formulaire actuel : COLLE LE CODE
Structure de stockage : COLLE CELLE DU PROMPT 15

Gère explicitement les trois cas, et montre-moi le code de chacun :
— ça marche : confirmation claire, formulaire vidé, la nouvelle donnée visible
— ça échoue : message honnête, saisie CONSERVÉE, possibilité de réessayer
— c'est en cours : état d'attente visible, bouton incliquable

Conserve la protection anti-double-envoi mise en place au prompt 14.
Dis-moi si ta modification l'affaiblit d'une quelconque façon.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Rappeler une protection déjà en place et demander si la nouvelle modification l'affaiblit est une habitude à prendre : c'est souvent en ajoutant une fonctionnalité qu'on casse silencieusement une sécurité posée la veille.

✓ **CE QUE TU DOIS OBTENIR**

Tu saisis, tu valides, tu fermes le navigateur, tu reviens : la donnée est toujours là. C'est le test qui prouve que le stockage fonctionne.

! **SI ÇA NE MARCHE PAS**

La donnée disparaît au rechargement ? Elle n'est pas enregistrée mais seulement affichée. Demande : « Montre-moi la ligne exacte qui écrit dans la base. »

18 Afficher la liste

NIVEAU : DÉBUTANT | APRÈS LE 17 | ~8 MIN

Quand t'en servir. Pour voir ce qui a été enregistré, et donner à l'utilisateur la preuve que son action a servi à quelque chose.

```
Crée un écran qui affiche tout ce qui a été enregistré, le plus récent en haut.

Traite explicitement ces trois situations :
— il n'y a encore rien : affiche un message qui explique quoi faire pour
  commencer, avec le bouton pour le faire. Jamais une page vide
— le chargement est en cours : montre-le, sans faire clignoter l'écran
— le chargement a échoué : dis-le, et propose un bouton « réessayer »

Affiche aussi le nombre total d'éléments.
Sur téléphone, chaque élément doit rester lisible sans zoomer.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

L'écran vide est le plus négligé et le plus décisif : c'est le premier que voit un nouvel utilisateur. Une page blanche donne l'impression que l'application est cassée ; un message qui dit quoi faire transforme ce vide en invitation.

✓ **CE QUE TU DOIS OBTENIR**

Trois états testables. Vide ta base pour voir l'écran vide : il doit t'inviter à agir, pas te laisser perplexe.

! **SI ÇA NE MARCHE PAS**

L'ordre est inversé ? Demande : « Trie par date de création décroissante, et montre-moi où se fait le tri. »

19 Modifier et supprimer

NIVEAU : INTERMÉDIAIRE | APRÈS LE 18 | ~10 MIN

Quand t'en servir. Dès que quelqu'un d'autre que toi utilise l'application. Les gens se trompent en saisissant : sans modification, ils t'écriront pour chaque faute de frappe.

```
Ajoute la modification et la suppression des éléments.

Pour la modification :
— le formulaire s'ouvre pré-rempli avec les valeurs actuelles
— annuler ne doit rien modifier
— enregistre la date de dernière modification

Pour la suppression, sois prudent :
— demande toujours confirmation
— la confirmation doit nommer précisément ce qui sera supprimé,
  pas « voulez-vous supprimer cet élément ? »
— dis-moi s'il vaut mieux marquer l'élément comme supprimé plutôt que
  l'effacer réellement, et quels sont les avantages de chaque approche
  dans mon cas
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Nommer ce qui va être supprimé dans la confirmation évite la moitié des suppressions accidentelles. Et la question sur la suppression douce est un vrai arbitrage de professionnel : effacer réellement libère de la place et respecte mieux la vie privée ; marquer permet de récupérer une erreur. Il n'y a pas de bonne réponse universelle — d'où la question.

✓ **CE QUE TU DOIS OBTENIR**

Une confirmation qui cite le nom de l'élément. Modifie un élément, recharge : la valeur modifiée est bien celle conservée.

! **SI ÇA NE MARCHE PAS**

La modification crée un doublon au lieu de remplacer ? Demande : « Mets à jour l'élément existant par son identifiant, n'en crée pas un nouveau. »

20 Chercher et filtrer

NIVEAU : INTERMÉDIAIRE | DÈS ~20 ÉLÉMENTS | ~8 MIN

Quand t'en servir. Quand la liste dépasse ce qui tient sur un écran. Avant, c'est du travail inutile.

```
Ajoute une recherche sur CE QUE TU VEUX POUVOIR CHERCHER
et un filtre par CRITÈRE.

Exigences :
— la recherche ignore les accents ET les majuscules : « ATIÉKÉ », « atieke »
  et « Atiéké » doivent donner le même résultat
— elle cherche dès la saisie, sans bouton, mais sans relancer à chaque lettre
  (attends une courte pause dans la frappe)
— affiche le nombre de résultats trouvés
— si aucun résultat : dis-le clairement et propose d'effacer la recherche
— la recherche et le filtre doivent pouvoir s'utiliser ensemble
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

L'insensibilité aux accents n'est pas un détail en français : sans elle, la moitié de tes utilisateurs ne trouveront rien, parce qu'on tape rarement les accents sur un téléphone. La pause dans la frappe évite de bombarder ta base à chaque lettre — ce qui compte quand la connexion est lente.

✓ **CE QUE TU DOIS OBTENIR**

Tape un mot avec et sans accent : même résultat. Tape n'importe quoi : un message clair, pas une liste vide sans explication.

! **SI ÇA NE MARCHE PAS**

Les accents bloquent ? Demande : « Normalise le texte et la recherche en retirant les accents avant de comparer. »

21 Ne jamais perdre une donnée

NIVEAU : EXIGEANT | AVANT LA MISE EN LIGNE | ~15 MIN

Quand t'en servir. Avant de laisser un vrai utilisateur entrer quoi que ce soit. Une donnée perdue est un client perdu, et tu ne sauras même pas que c'est arrivé.

```
Relis tout mon code et identifie chaque situation où une donnée saisie
par un utilisateur pourrait être perdue.

Examine au minimum :
— la connexion qui coupe pendant l'envoi
— le serveur ou la base qui répond une erreur
— l'utilisateur qui ferme l'onglet pendant l'enregistrement
— deux personnes qui modifient le même élément en même temps
— un envoi qui réussit mais dont la confirmation n'arrive jamais

Pour chaque cas : où ça se produit dans mon code, ce que perd l'utilisateur,
et la correction la PLUS SIMPLE possible.
Classe-les du plus grave au moins grave, et dis-moi lesquels je peux
raisonnablement ignorer pour une première version.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Le cas « l'envoi réussit mais la confirmation n'arrive jamais » est celui auquel personne ne pense, et c'est celui qui crée les doublons en conditions de réseau instable — fréquent en mobile. Demander un classement et une liste d'« ignorables » évite de te noyer : tu corriges les trois premiers et tu avances.

✓ **CE QUE TU DOIS OBTENIR**

Une liste classée avec au moins un cas que tu n'avais pas envisagé. Corrige les deux ou trois premiers, note les autres pour plus tard.

! **SI ÇA NE MARCHE PAS**

L'IA te dit que tout va bien ? Insiste : « Fais l'hypothèse que mon code a des failles. Trouve-en trois. » Elle en trouvera.

PHASE 4 — LES COMPTES UTILISATEURS

> À n'ajouter que si ton application en a réellement besoin. Beaucoup de premiers projets fonctionnent très bien sans — et chaque compte est une friction de plus à l'entrée.

22 Décider si tu en as vraiment besoin

NIVEAU : DÉBUTANT | AVANT TOUT LE RESTE | ~4 MIN

Quand t'en servir. Avant de te lancer dans ce qui est, de loin, la partie la plus longue d'une application. Une réponse honnête ici peut t'économiser une semaine.

```
Voici ce que fait mon application : DÉCRIS-LA
Voici qui l'utilise : COLLE LE PROMPT 04

Ai-je vraiment besoin de comptes utilisateurs pour la première version,
ou puis-je m'en passer ?

Réponds franchement, en examinant :
— ce que je perds concrètement sans comptes
— ce que ça me coûte en temps de les ajouter
— s'il existe une solution intermédiaire (un simple lien unique, un code,
  une identification par numéro de téléphone)

Si je peux m'en passer, dis-le clairement, même si c'est moins impressionnant.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

La dernière ligne est la plus importante : sans elle, un modèle a tendance à proposer la solution la plus complète plutôt que la plus adaptée. L'option « lien unique ou code » est souvent la bonne réponse pour une première version, et presque personne n'y pense.

✓ **CE QUE TU DOIS OBTENIR**

Une réponse tranchée, pas un « ça dépend ». Si c'est non, saute directement à la phase 5 — tu viens de gagner plusieurs jours.

! **SI ÇA NE MARCHE PAS**

Réponse évasive ? Reformule : « Décide à ma place, et assume. Oui ou non ? »

23 Inscription et connexion

NIVEAU : EXIGEANT | SEULEMENT SI LE 22 DIT OUI | ~25 MIN

Quand t'en servir. Une fois la décision prise. C'est la partie où il ne faut surtout pas improviser : la sécurité des mots de passe n'est pas un terrain d'expérimentation.

```
Ajoute l'inscription et la connexion à mon application.

Contraintes non négociables :
— je ne veux JAMAIS stocker ni gérer un mot de passe moi-même :
  utilise un service éprouvé qui s'en charge
— choisis la solution gratuite la plus simple à mettre en route
— explique-moi où sont stockés les comptes et comment je les consulte

Donne-moi :
1. les étapes de configuration, numérotées, sans vocabulaire technique
2. le code de l'inscription et de la connexion
3. les messages d'erreur en français, sans jamais révéler si c'est l'e-mail
   ou le mot de passe qui est faux
4. ce que je dois tester moi-même pour être sûr que ça fonctionne

Explique-moi aussi, en une phrase, pourquoi le point 3 est important.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Ne jamais gérer un mot de passe soi-même » est la règle la plus importante de cette phase : stocker des mots de passe correctement demande une expertise que ni toi ni l'IA n'avez à improviser. Le point 3 — ne pas révéler lequel des deux champs est faux — évite qu'un inconnu puisse deviner quelles adresses existent chez toi.

✓ **CE QUE TU DOIS OBTENIR**

Tu crées un compte, tu te déconnectes, tu te reconnectes. Puis tu essaies un mauvais mot de passe : le message ne doit pas dire que l'adresse existe.

! **SI ÇA NE MARCHE PAS**

Si l'IA propose de stocker elle-même les mots de passe, refuse net : « Non. Utilise un service d'authentification existant. »

24 Protéger les pages privées

NIVEAU : EXIGEANT | IMMÉDIATEMENT APRÈS LE 23 | ~12 MIN

Quand t'en servir. Dans la foulée de la connexion, jamais « plus tard ». Une page privée non protégée est une fuite de données qui ne prévient pas.

```
Rends ces pages inaccessibles à quelqu'un qui n'est pas connecté :
LISTE LES PAGES CONCERNÉES

Exigences :
— un visiteur non connecté est renvoyé vers la page de connexion
— après connexion, il revient sur la page qu'il voulait voir au départ
— la vérification se fait AUSSI côté serveur, pas seulement dans le navigateur

Puis, très important : montre-moi précisément la ligne de code qui fait
la vérification côté serveur, et explique-moi comment quelqu'un pourrait
contourner une protection qui n'existerait que côté navigateur.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Une protection présente uniquement dans le navigateur se contourne en quelques secondes, avec les outils intégrés à n'importe quel navigateur — ce n'est pas du piratage, c'est un menu. Demander à l'IA de te montrer la ligne serveur *et* d'expliquer le contournement t'oblige à voir la différence de tes propres yeux.

✓ **CE QUE TU DOIS OBTENIR**

Déconnecte-toi, colle l'adresse d'une page privée directement dans la barre du navigateur : tu dois être renvoyé vers la connexion.

! **SI ÇA NE MARCHE PAS**

La page s'affiche brièvement avant la redirection ? Les données ont déjà fuité. Demande : « Vérifie l'autorisation AVANT d'envoyer le contenu, pas après. »

25 Des rôles différents

NIVEAU : EXIGEANT | SI TOUT LE MONDE NE VOIT PAS LA MÊME CHOSE | ~12 MIN

Quand t'en servir. Quand il y a un administrateur et des utilisateurs, ou un vendeur et des clients. Pas avant.

```
Ajoute deux rôles à mon application : DÉCRIS-LES
(par exemple : administrateur et client)

Pour chaque rôle, établis d'abord un tableau clair :
ce qu'il peut VOIR, ce qu'il peut CRÉER, ce qu'il peut MODIFIER,
ce qu'il peut SUPPRIMER.

Puis applique ces règles :
— la vérification se fait côté serveur pour CHAQUE action, pas seulement
  en masquant des boutons à l'affichage
— masquer un bouton n'est pas une protection, c'est du confort

Montre-moi une action où un utilisateur normal pourrait, en s'y prenant mal
intentionnellement, faire quelque chose réservé à l'administrateur —
et comment tu l'en empêches.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Le tableau avant le code force à décider des règles avant de les implémenter — sinon elles se découvrent en chemin, incohérentes. Et demander à l'IA de chercher elle-même une faille dans ce qu'elle vient de proposer est une technique simple et redoutablement efficace.

✓ **CE QUE TU DOIS OBTENIR**

Un tableau des permissions que tu peux lire d'un coup d'œil, et au moins une faille identifiée puis fermée.

! **SI ÇA NE MARCHE PAS**

La protection n'est que visuelle ? Demande : « Ajoute la vérification du rôle côté serveur sur chaque action sensible, et liste-les. »

26 Mot de passe oublié

NIVEAU : INTERMÉDIAIRE | AVANT LA MISE EN LIGNE | ~10 MIN

Quand t'en servir. Avant d'ouvrir à de vrais utilisateurs. Sans cette fonction, tu réinitialiseras les mots de passe à la main, un par un, par message.

```
Ajoute la réinitialisation du mot de passe par e-mail.

Exigences :
— le lien envoyé expire après un délai court, et ne fonctionne qu'une seule fois
— le message affiché est le même que l'adresse existe ou non
  (« si cette adresse est connue, tu recevras un lien »)
— tous les textes en français, simples et rassurants
— une fois le mot de passe changé, les sessions ouvertes ailleurs sont fermées

Explique-moi pourquoi le deuxième point compte.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Le message identique quelle que soit l'adresse empêche un inconnu de tester des milliers d'adresses pour découvrir lesquelles ont un compte chez toi. Fermer les autres sessions protège le cas où quelqu'un réinitialise justement parce qu'il pense être piraté.

✓ **CE QUE TU DOIS OBTENIR**

Teste le parcours complet avec ta propre adresse, puis réutilise le même lien : il doit être refusé la deuxième fois.

! **SI ÇA NE MARCHE PAS**

Le mail n'arrive pas ? Regarde dans les indésirables avant de toucher au code — c'est la cause la plus fréquente.

27 L'écran de profil

NIVEAU : INTERMÉDIAIRE | QUAND ON TE LE DEMANDE | ~8 MIN

Quand t'en servir. Quand tes utilisateurs commencent à te demander de corriger leurs informations à leur place.

```
Crée un écran où l'utilisateur voit et modifie ses propres informations.

Exigences :
— un changement d'adresse e-mail doit être confirmé par un lien envoyé
  à la NOUVELLE adresse ; tant que ce n'est pas confirmé, l'ancienne reste active
— un changement de mot de passe exige de saisir l'ancien
— prévois un moyen pour l'utilisateur de supprimer son compte,
  et dis-moi ce qui doit être effacé et ce qui doit être conservé
— confirme visiblement chaque enregistrement réussi
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Confirmer sur la nouvelle adresse évite qu'une faute de frappe rende un compte inaccessible à jamais. Exiger l'ancien mot de passe empêche qu'un ordinateur laissé ouvert permette à un tiers de prendre le contrôle du compte.

✓ **CE QUE TU DOIS OBTENIR**

Change ton adresse pour une faute volontaire : ton ancienne adresse doit continuer de fonctionner tant que tu n'as pas confirmé.

! **SI ÇA NE MARCHE PAS**

Le changement est immédiat sans confirmation ? Demande : « N'applique le nouvel e-mail qu'après clic sur le lien de confirmation. »

PHASE 5 — ENCAISSER EN FCFA

> Le moment où l'application cesse d'être un projet et devient une affaire. C'est aussi là qu'on commet les erreurs les plus coûteuses. Les cinq fiches qui suivent viennent de projets réels, en production, qui encaissent tous les jours.

28 Choisir comment encaisser

NIVEAU : INTERMÉDIAIRE | AVANT TOUTE LIGNE DE CODE | ~15 MIN

Quand t'en servir. Avant d'écrire quoi que ce soit lié au paiement. Le choix de l'intermédiaire conditionne tout le reste, et en changer plus tard coûte cher.

```
Je veux encaisser des paiements en FCFA, principalement par mobile money,
depuis PAYS, auprès de clients situés en PAYS OU RÉGION.

Compare les solutions disponibles sur ces critères précis :
— quels opérateurs mobile money elles couvrent réellement dans mon pays
— acceptent-elles aussi les cartes bancaires
— quelle commission par transaction
— quel délai avant que l'argent arrive sur mon compte
— quels documents sont exigés pour ouvrir un compte professionnel
— la qualité de leur documentation technique pour un débutant

Présente le résultat en tableau, puis recommande-m'en une pour mon cas
précis, en assumant ton choix.

Préviens-moi si une information peut avoir changé récemment et mérite
d'être vérifiée directement chez l'opérateur.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Le délai de versement est le critère que tout le monde oublie et qui compte le plus quand on démarre : encaisser 50 000 F qui n'arrivent que trois semaines plus tard change ta trésorerie. La dernière ligne est une précaution sérieuse : les tarifs et couvertures des opérateurs bougent, et un modèle peut travailler sur des informations dépassées.

✓ **CE QUE TU DOIS OBTENIR**

Un tableau comparatif et une recommandation nette. Vérifie ensuite les commissions directement sur le site de l'opérateur retenu avant de t'engager.

! **SI ÇA NE MARCHE PAS**

La comparaison reste vague ? Demande : « Pour chaque solution, donne-moi le nom exact des opérateurs mobile money couverts dans mon pays. »

29 Brancher le paiement

NIVEAU : EXIGEANT | COMPTE OPÉRATEUR OUVERT | ~30 MIN

Quand t'en servir. Une fois ton compte validé chez l'intermédiaire, avec tes clés en main. Prends ce prompt au calme, pas à minuit.

```
Branche SOLUTION CHOISIE sur mon application, pour encaisser un paiement
de MONTANT FCFA correspondant à CE QUE TU VENDS.

Règle absolue :
mes clés secrètes ne doivent JAMAIS se retrouver dans le code exécuté par
le navigateur, ni dans un fichier publié. Explique-moi exactement où les
placer pour qu'elles restent côté serveur, et comment vérifier moi-même
qu'elles n'ont pas fuité.

Donne-moi :
1. les étapes de configuration chez l'opérateur, numérotées
2. le code du déclenchement du paiement
3. la marche à suivre pour tester avec un VRAI paiement du plus petit
   montant possible, avec mon propre téléphone
4. comment retrouver une transaction dans le tableau de bord de l'opérateur

Distingue clairement les clés de test et les clés de production,
et dis-moi comment je reconnais laquelle j'utilise.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Une clé secrète visible dans le navigateur, c'est ton compte vidé — et ça arrive par simple inattention, pas par piratage sophistiqué. Le point 3 impose un vrai paiement de test : les paiements mobile money se comportent différemment en conditions réelles, et beaucoup de surprises n'apparaissent qu'avec un vrai téléphone.

✓ **CE QUE TU DOIS OBTENIR**

Un paiement réel du plus petit montant possible, reçu, et retrouvable dans le tableau de bord de l'opérateur. Tant que ce test n'est pas fait, considère que rien ne marche.

! **SI ÇA NE MARCHE PAS**

Rien ne se déclenche sur le téléphone ? Vérifie d'abord que tu es en clés de production et non en clés de test — c'est la confusion numéro un.

30 Le montant minimum

NIVEAU : INTERMÉDIAIRE | AVANT LE PREMIER VRAI CLIENT | ~6 MIN

Quand t'en servir. Juste après avoir branché le paiement. C'est un détail invisible qui fait échouer des transactions sans que tu comprennes pourquoi.

```
Quel est le montant minimum accepté par SOLUTION CHOISIE et par les
opérateurs mobile money de mon pays ?

Ajoute ensuite à mon application une vérification qui :
— refuse un montant inférieur AVANT même de lancer la demande de paiement
— affiche un message qui explique pourquoi et indique le minimum
— ne laisse jamais l'utilisateur découvrir le problème après avoir reçu
  une demande de paiement sur son téléphone

Prévois aussi le cas inverse : un montant anormalement élevé,
probablement dû à une erreur de saisie. Demande une confirmation.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Beaucoup d'opérateurs refusent en dessous d'un certain seuil, souvent autour de 100 FCFA. Si tu vends un service à très petit prix — un vote, une contribution, un accès ponctuel — tu dois le savoir avant de lancer, pas après quinze échecs inexpliqués. La confirmation sur montant élevé t'évite un remboursement gênant.

✓ **CE QUE TU DOIS OBTENIR**

Saisis un montant sous le seuil : le refus doit arriver immédiatement, avec le minimum indiqué, sans qu'aucune demande ne parte sur ton téléphone.

! **SI ÇA NE MARCHE PAS**

La demande part quand même ? La vérification est faite trop tard. Demande : « Vérifie le montant avant d'appeler l'opérateur, pas après. »

31 Ne jamais re-router un paiement lancé

NIVEAU : EXIGEANT | AVANT TOUT CLIENT RÉEL | ~12 MIN

Quand t'en servir. Avant le premier vrai client. C'est la règle la plus chère à apprendre sur le terrain, et celle que personne ne t'expliquera avant que ça t'arrive.

```
Voici la règle à appliquer dans mon application :

Une fois que la demande de paiement est partie vers le téléphone du client
— c'est-à-dire dès qu'il a reçu la notification de validation —
mon application ne doit JAMAIS :
— relancer la même demande
— la rediriger vers un autre opérateur
— en créer une deuxième parce que le client a recliqué
— réessayer automatiquement en cas de non-réponse

Mets en place ce verrou. Puis :
1. montre-moi précisément où il agit dans le code
2. explique-moi ce que voit l'utilisateur pendant l'attente
3. dis-moi comment on sort proprement d'une demande restée sans réponse
   (expiration, annulation manuelle, vérification du statut)

Explique-moi aussi ce qui arriverait, concrètement, sans ce verrou.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Sans ce verrou, un client impatient qui reclique, ou un système qui « réessaie pour bien faire », déclenche une deuxième demande. Si le client valide les deux, il est débité deux fois — et rembourser en mobile money est long, manuel et humiliant. C'est la panne la plus coûteuse qu'on puisse s'infliger, et elle vient toujours d'une bonne intention : vouloir réessayer.

✓ **CE QUE TU DOIS OBTENIR**

Lance un paiement, puis reclique frénétiquement sur le bouton : une seule demande doit arriver sur ton téléphone. Fais ce test pour de vrai.

! **SI ÇA NE MARCHE PAS**

Deux notifications arrivent ? Demande : « Bloque toute nouvelle demande tant que la précédente n'est ni expirée ni conclue, et montre-moi le verrou. »

32 Confirmer et donner l'accès

NIVEAU : EXIGEANT | DERNIÈRE ÉTAPE DU PAIEMENT | ~15 MIN

Quand t'en servir. Pour fermer la boucle : le client a payé, il doit recevoir ce qu'il a acheté, automatiquement et une seule fois.

```
Quand un paiement est confirmé, mon application doit donner l'accès
automatiquement à CE QUE LE CLIENT A ACHETÉ.

Trois règles impératives :
1. ne te fie JAMAIS au retour affiché dans le navigateur pour considérer
   qu'un paiement est validé : il peut être falsifié. Attends la confirmation
   envoyée directement par l'opérateur à mon serveur
2. cette confirmation peut arriver DEUX FOIS, ou en double quelques secondes
   plus tard : l'accès ne doit être donné qu'une seule fois
3. elle peut aussi arriver en retard, après que le client a fermé la page :
   ça doit marcher quand même

Montre-moi le code de chacune des trois règles.
Puis dis-moi comment je vérifie moi-même, dans le tableau de bord de
l'opérateur, qu'un paiement donné a bien été traité une seule fois.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Se fier au retour du navigateur, c'est laisser n'importe qui s'offrir ton produit en modifiant une adresse. Et les opérateurs renvoient fréquemment la même confirmation plusieurs fois, par sécurité : si ton code ne le prévoit pas, un client reçoit deux accès, ou tu comptes deux ventes pour une.

✓ **CE QUE TU DOIS OBTENIR**

Un vrai paiement de test qui donne l'accès automatiquement. Ferme la page juste après avoir validé sur ton téléphone : l'accès doit quand même être accordé.

! **SI ÇA NE MARCHE PAS**

L'accès n'arrive que si on reste sur la page ? La confirmation est traitée côté navigateur. Demande : « Traite la confirmation côté serveur, indépendamment du navigateur du client. »

PHASE 6 — METTRE EN LIGNE

> Tant que ça tourne sur ton ordinateur, ça n'existe pas. Quatre prompts pour obtenir une adresse que tu peux envoyer à quelqu'un.

33 Préparer la mise en ligne

NIVEAU : EXIGEANT | AVANT DE PUBLIER | ~12 MIN

Quand t'en servir. Juste avant de publier. C'est le dernier moment où une clé secrète peut encore être retirée avant de devenir publique pour toujours.

```
Prépare mon projet à être mis en ligne.

Vérifie et retire, en me listant tout ce que tu enlèves et pourquoi :
— toute clé secrète, mot de passe ou identifiant de connexion
— toute donnée de test, faux client, faux paiement
— tout commentaire contenant une information sensible
— tout accès de débogage laissé ouvert
— toute page ou fichier qui ne devrait pas être accessible publiquement

Puis donne-moi la liste des fichiers qui seront réellement publiés,
et confirme qu'aucun fichier sensible n'en fait partie.

Dis-moi enfin comment je peux vérifier moi-même, une fois en ligne,
qu'un fichier sensible n'est pas accessible.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Demander la liste des fichiers *réellement publiés* révèle les oublis : un fichier de configuration, une sauvegarde, un export de données laissé dans le dossier. Et savoir le vérifier soi-même après coup compte plus que la promesse : il suffit d'essayer d'ouvrir l'adresse du fichier pour voir.

✓ **CE QUE TU DOIS OBTENIR**

Une liste explicite de ce qui part et de ce qui reste. Après publication, essaie d'ouvrir l'adresse d'un fichier sensible : tu dois obtenir une erreur.

! **SI ÇA NE MARCHE PAS**

Un fichier sensible est accessible ? Retire-le immédiatement, publie à nouveau, puis change les clés qu'il contenait. Elles doivent être considérées comme compromises.

34 Publier

NIVEAU : INTERMÉDIAIRE | APRÈS LE 33 | ~20 MIN

Quand t'en servir. Le grand moment. Après ça, ton projet a une adresse.

```
Donne-moi la façon la plus simple et gratuite de mettre ce projet en ligne.

Contraintes :
— je n'ai jamais publié de site de ma vie
— je veux une adresse que je peux envoyer à quelqu'un aujourd'hui
— gratuit, ou au moins gratuit pour commencer

Donne-moi :
1. le service retenu et pourquoi il convient à mon cas
2. les étapes numérotées, en supposant que je ne connais aucun terme technique
3. où configurer mes clés secrètes pour qu'elles ne soient pas publiées
4. à quoi je reconnaîtrai précisément que ça a marché
5. combien de temps il faut attendre avant que l'adresse fonctionne

Préviens-moi de ce qui coûtera de l'argent plus tard, et à partir de quand.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

La dernière ligne évite la mauvaise surprise du service gratuit qui devient payant dès que tu as un peu de trafic. Le point 5 t'évite de croire que tu as raté quelque chose alors qu'il faut simplement attendre quelques minutes.

✓ **CE QUE TU DOIS OBTENIR**

Une adresse que tu ouvres sur ton téléphone, en données mobiles, hors de ton réseau habituel. Si ça s'affiche, c'est en ligne.

! **SI ÇA NE MARCHE PAS**

La publication échoue ? Copie le message complet et passe au prompt 37 : les erreurs de publication sont généralement très explicites.

35 Mettre ton nom de domaine

NIVEAU : INTERMÉDIAIRE | QUAND TU VEUX UNE ADRESSE À TOI | ~15 MIN + ATTENTE

Quand t'en servir. Une fois le site en ligne et stable. Avant, tu configurerais un domaine vers quelque chose qui bouge encore.

```
J'ai acheté le domaine MONDOMAINE.COM chez NOM DU VENDEUR.
Mon site est publié chez SERVICE DU PROMPT 34.

Explique-moi exactement quoi configurer pour que mon domaine pointe
vers mon application :
— chez qui je fais la manipulation
— quelles lignes exactes je dois ajouter, avec leurs valeurs
— dans quel ordre

Puis dis-moi :
— combien de temps avant que ça fonctionne, et pourquoi ce n'est pas immédiat
— comment vérifier que c'est en cours plutôt que cassé
— si le site restera accessible par l'ancienne adresse pendant la transition
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Pourquoi ce n'est pas immédiat » évite la panique légitime du débutant qui reconfigure tout trois fois en dix minutes alors qu'il suffisait d'attendre. Savoir distinguer « en cours » de « cassé » est la seule information qui compte pendant ce délai.

✓ **CE QUE TU DOIS OBTENIR**

Des lignes de configuration précises et un délai annoncé. Attends ce délai avant de conclure quoi que ce soit.

! **SI ÇA NE MARCHE PAS**

Après le délai annoncé, toujours rien ? Demande : « Donne-moi une commande ou un site pour vérifier où pointe actuellement mon domaine. »

36 Vérifier comme un vrai utilisateur

NIVEAU : DÉBUTANT | JUSTE APRÈS LA PUBLICATION | ~20 MIN

Quand t'en servir. Immédiatement après la mise en ligne, et avant d'envoyer l'adresse à qui que ce soit. Ce que tu testes sur ton ordinateur ne prouve rien.

```
Donne-moi une liste de vérifications à effectuer moi-même sur le site
maintenant qu'il est en ligne, depuis un téléphone, comme un vrai utilisateur.

Couvre :
— le parcours complet, du début à la fin, sans rien sauter
— ce qui se passe avec une connexion lente ou instable
— ce qui se passe si on ferme la page en plein milieu
— les cas d'erreur : mauvaise saisie, paiement annulé, double clic
— l'affichage sur un petit écran

Présente ça comme une liste de cases à cocher, une ligne par vérification,
formulée à l'impératif et sans vocabulaire technique.
Mets en premier les vérifications qui, si elles échouent, rendent
l'application inutilisable.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Une liste ordonnée par gravité te permet d'arrêter dès qu'un point critique échoue, au lieu de dérouler trente vérifications pour découvrir à la fin que rien ne marche. Le format « cases à cocher, à l'impératif » en fait un outil réutilisable à chaque mise à jour.

✓ **CE QUE TU DOIS OBTENIR**

Une liste que tu peux dérouler en vingt minutes sur ton téléphone. Fais-la entièrement, même les points qui te paraissent évidents.

! **SI ÇA NE MARCHE PAS**

Un point échoue ? Note-le, termine la liste, puis corrige par ordre de gravité. Ne corrige pas au fil de l'eau : tu perdrais le fil.

PHASE 7 — QUAND ÇA CASSE

> Ça cassera. Ce n'est pas le signe que tu n'y arrives pas : c'est le métier. Ces quatre prompts valent tous les autres réunis, parce que ce sont eux qui t'évitent d'abandonner.

37 Faire expliquer l'erreur

NIVEAU : DÉBUTANT | DEVANT UN MESSAGE INCOMPRÉHENSIBLE | ~5 MIN

Quand t'en servir. Dès qu'un message d'erreur apparaît. Avant toute tentative de correction.

```
J'ai cette erreur :
COLLE LE MESSAGE COMPLET, SANS RIEN COUPER

Elle survient quand : DÉCRIS L'ACTION EXACTE QUI LA DÉCLENCHE

Ne corrige rien pour l'instant. Je veux d'abord comprendre.

Explique-moi :
1. ce que ce message signifie, en français simple
2. ce qui le provoque habituellement
3. les trois causes les plus probables dans MON cas précis,
   de la plus probable à la moins probable
4. pour chaque cause, comment la vérifier en une seule manipulation

Si le message ne suffit pas pour trancher, dis-moi quelle information
supplémentaire te serait utile.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Ne corrige rien pour l'instant » est la consigne décisive. Sans elle, le modèle propose immédiatement une correction, tu l'appliques, et si elle est fausse tu as maintenant deux problèmes. Comprendre avant de réparer est ce qui te rendra autonome.

✓ **CE QUE TU DOIS OBTENIR**

Une explication que tu comprends et trois pistes vérifiables. Vérifie-les dans l'ordre, une à la fois.

! **SI ÇA NE MARCHE PAS**

L'explication reste obscure ? Demande : « Explique-le comme à quelqu'un qui n'a jamais programmé, avec une comparaison du quotidien. »

38 Isoler le problème

NIVEAU : INTERMÉDIAIRE | QUAND IL N'Y A AUCUN MESSAGE | ~10 MIN

Quand t'en servir. Quand ça ne marche pas mais que rien ne s'affiche. Le cas le plus déroutant, et celui où on perd le plus de temps.

```
Voici ce qui DEVRAIT se passer : DÉCRIS
Voici ce qui se passe VRAIMENT : DÉCRIS PRÉCISÉMENT, SANS INTERPRÉTER

Aucun message d'erreur n'apparaît.

Propose-moi une série de vérifications pour trouver d'où vient le problème :
— de la plus simple et rapide à la plus longue
— UNE SEULE à la fois
— pour chacune : quoi faire exactement, et quoi conclure selon le résultat

Après chaque étape, attends que je te donne le résultat avant de passer
à la suivante. Ne me donne pas toute la liste d'un coup.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Demander une seule vérification à la fois transforme le dépannage en dialogue plutôt qu'en liste illisible. C'est exactement la méthode d'un développeur expérimenté : réduire le champ des possibles par moitié à chaque test. « Sans interpréter » est important aussi : décris ce que tu vois, pas ce que tu crois comprendre.

✓ **CE QUE TU DOIS OBTENIR**

Une première vérification simple, à faire tout de suite. Réponds avec le résultat exact et laisse le dialogue se dérouler.

! **SI ÇA NE MARCHE PAS**

Toute la liste arrive d'un coup ? Réponds : « Une seule à la fois. Commence par la première et attends ma réponse. »

39 Revenir en arrière

NIVEAU : DÉBUTANT | QUAND ÇA MARCHAIT IL Y A DIX MINUTES | ~5 MIN

Quand t'en servir. Quand une modification a tout cassé et que tu ne sais plus dans quel état tu es. Le réflexe qui sauve les soirées.

```
Ça fonctionnait avant cette modification :
COLLE CE QUI A ÉTÉ MODIFIÉ, OU DÉCRIS CE QUE TU AS DEMANDÉ JUSTE AVANT

Annule UNIQUEMENT cette modification et remets le code dans son état
précédent. Ne touche à rien d'autre, même si tu vois autre chose
à améliorer.

Ensuite, explique-moi :
— pourquoi cette modification cassait quelque chose
— comment on aurait pu obtenir le même résultat sans casser
— à quoi j'aurais pu le voir plus tôt
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

« Ne touche à rien d'autre, même si tu vois autre chose à améliorer » est nécessaire : les modèles aiment améliorer au passage, et c'est exactement ce qu'il ne faut pas quand tu cherches à revenir à un état connu. Les trois questions finales transforment l'incident en apprentissage.

✓ **CE QUE TU DOIS OBTENIR**

Le code revenu à son état fonctionnel, et une explication qui t'évitera de refaire la même erreur.

! **SI ÇA NE MARCHE PAS**

Tu ne sais plus ce qui a changé ? C'est le signal qu'il te faut des sauvegardes. Duplique ton dossier avant chaque grosse modification — c'est rudimentaire, mais ça marche.

40 Faire relire ton application

NIVEAU : EXIGEANT | AVANT CHAQUE MISE EN LIGNE | ~20 MIN

Quand t'en servir. Avant de montrer l'application, et avant chaque mise à jour importante. C'est la relecture que personne ne fait, et c'est celle qui évite les catastrophes.

```
Relis l'intégralité de mon projet comme le ferait un développeur
expérimenté chargé d'une revue avant mise en production.

Signale-moi UNIQUEMENT les problèmes sérieux :
— une donnée utilisateur qui peut être perdue
— une clé, un mot de passe ou un identifiant visible publiquement
— une page ou une action privée accessible sans être connecté
— un paiement qui pourrait être compté ou encaissé deux fois
— une saisie utilisateur utilisée sans être vérifiée

Ignore le style, les préférences et les optimisations.

Pour chaque problème trouvé :
— où exactement (fichier et ligne)
— pourquoi c'est grave, concrètement, avec un exemple de ce qui peut arriver
— la correction la plus simple possible

Classe du plus grave au moins grave.
Si tu ne trouves rien de sérieux, dis-le clairement plutôt que d'inventer
des remarques mineures.
```

◆ **POURQUOI IL EST ÉCRIT COMME ÇA**

Restreindre à cinq catégories précises évite de recevoir cinquante remarques de style qui noieraient les deux vraies failles. Et l'autorisation finale — « dis-le si tu ne trouves rien » — est indispensable : sans elle, un modèle inventera des problèmes pour paraître utile, et tu perdras ta confiance dans l'exercice.

✓ **CE QUE TU DOIS OBTENIR**

Une liste courte et classée. Corrige les deux ou trois premiers avant de publier. Relance ce prompt après chaque correction.

! **SI ÇA NE MARCHE PAS**

Tu reçois des remarques de style ? Réponds : « Ignore le style. Uniquement les cinq catégories que j'ai listées. »

ET MAINTENANT ?

Si tu as suivi ces quarante prompts, tu as une application qui tourne et une adresse à donner. C'est déjà plus loin que l'immense majorité des gens qui ont « une idée d'application » — la plupart n'ont jamais dépassé la phase 1.

LES CINQ ERREURS QUI FONT PERDRE LE PLUS DE TEMPS

- **Vouloir tout construire d'un coup.** Le prompt 02 existe pour ça. Quand tu bloques, c'est presque toujours qu'il faut y revenir.
- **Passer à la suite sans avoir testé.** Dix prompts plus tard, tu ne sauras plus lequel a introduit le problème.
- **Laisser la mise en page pour la fin.** Tes utilisateurs sont sur téléphone. Une application illisible est une application inutilisée.
- **Brancher les paiements sans les tester en vrai.** Fais un vrai paiement, du plus petit montant possible, avec ton propre téléphone. Toujours.
- **Croire l'IA sur parole.** Elle se trompe avec beaucoup d'assurance. Les prompts 37 à 40 sont tes garde-fous — utilise-les.

CE QUE CE GUIDE NE T'A PAS DONNÉ

Il t'a donné les prompts et la méthode pour les enchaîner. Il ne t'a pas donné **la plomberie déjà faite** : la structure de projet éprouvée, les comptes utilisateurs, les paiements FCFA déjà branchés et testés. Ni l'ordre exact dans lequel je procède sur un vrai projet client, qui n'est pas tout à fait celui-ci.

LA SUITE, SI TU VEUX ALLER PLUS VITE

C'est ce qu'il y a dans **« Ton 1er SaaS »** : la méthode complète pas à pas, et le kit boilerplate offert avec — celui qui t'évite les trois semaines de plomberie dont tout le monde parle trop tard.

Voir « Ton 1er SaaS » →

masterclass.dagbegnonagency.com/offre

UNE QUESTION, UN BLOCAGE ?

Écris-moi directement sur WhatsApp au **+229 61 14 92 58**. C'est moi qui réponds, et je lis tout. Dis-moi surtout *où* tu as bloqué et sur quel numéro de prompt : c'est exactement ce qui me sert à améliorer ce guide pour les suivants.

Ce document t'est offert. Tu peux le partager tel quel, sans le modifier.