



CAHIER DES CHARGES

PRESS AI

**Application mobile de transformation d'interviews
audio en articles journalistiques par IA**

Projet : BUT Informatique 2e - Développement mobile avec FlutterFlow

Auteur : Grégoire MAROUSE BUT 2B1

Date : Décembre 2025

Press AI est une application mobile innovante qui révolutionne le travail des journalistes en transformant automatiquement des enregistrements audio d'interviews en articles professionnels structurés.

Le problème

Les journalistes et créateurs de contenu passent en moyenne 4 à 6 heures à retranscrire et rédiger un article à partir d'une interview d'une heure. Ce processus répétitif et chronophage limite leur productivité. Plus spécifiquement pour les correspondants locaux le temps passé sur place additionné au temps de rédaction est beaucoup trop important pour des articles payés au lance pierre. L'objectif est de concentrer leur activité dans les relations avec les intervenants et que la rédaction se fasse en automatique. Pour réduire le temps nécessaire à la production d'un article.

La solution

Press AI automatise ce processus grâce à l'intelligence artificielle :

1. L'utilisateur sélectionne un fichier audio sur son appareil
2. L'audio est transcrit automatiquement (API de transcription)
3. Un article structuré est généré par IA (Agent IA rédactionnel)
4. L'utilisateur personnalise et édite l'article
5. L'article est exporté au format souhaité

Gain de temps cible

80% de réduction du temps de production : de 4-6h à moins d'1h par article.

Architecture simplifiée

- **Frontend** : FlutterFlow (développement mobile low-code)
- **Authentification** : Firebase Authentification uniquement
- **APIs** : Transcription (à définir)
- **Agent IA** : Agent rédactionnel basé sur un modèle à définir
- **Stockage** : AUCUN - Application stateless (sans sauvegarde cloud)
- **Export** : Fichiers locaux (.txt, .md, partage système)

Contraintes

- Budget : 0€ (free tiers uniquement)
- Durée développement : 10 heures de cours (+ temps perso)
- Aucun stockage cloud (pas de coûts récurrents)
- Interface en français
- Utilisation de login avec Firebase
- utilisation d'API
- custom actions / custom functions
- page de configuration (stocké en app state)

TABLE DES MATIÈRES

CONTEXTE DU PROJET

Problématique

Objectifs

Objectifs fonctionnels

Objectifs pédagogiques

Objectifs techniques

Périmètre et limites

INCLUS dans la v1

EXCLU de la v1

PALETTE DE COULEURS

SPÉCIFICATION FONCTIONNELLES

Diagramme des cas d'usages

Exemple d'utilisation par un utilisateur

Diagramme d'activité

Map des interfaces du site

Wireframe des interfaces

SPÉCIFICATION TECHNIQUES

1. ARCHITECTURE DE DONNÉES

1.1 Base de données - Firebase Firestore

2. INTÉGRATIONS API EXTERNES

2.1 Groq API - Transcription audio (Whisper Large v3)

2.2 Groq API - Génération d'articles (Llama 3.3 70B)

3. CUSTOM ACTIONS & CUSTOM FUNCTIONS

3.1 Custom Actions

3.2 Custom Functions

4. SYSTÈME D'AUTHENTIFICATION

5. PAGES & NAVIGATION

6. GESTION D'ÉTAT - AppState Variables

7. CONTRAINTES TECHNIQUES

7.1 Plateformes de déploiement

7.2 Responsive Design

7.3 Limitations techniques

8. SÉCURITÉ & GESTION DES ERREURS

8.1 Sécurité Firestore

8.2 Gestion des erreurs

9. TESTS & VALIDATION

9.1 Méthodologie de test

GESTION DE PROJET

Diagramme de Gantt

CONTEXTE DU PROJET

Problématique

Les journalistes, communicants et créateurs de contenu font face à un défi majeur : **la retranscription et la rédaction d'interviews prennent un temps considérable.**

Constat chiffré :

- Retranscription manuelle : **4x la durée de l'audio** (1h d'interview = 4h de travail)
- Structuration en article : **2h supplémentaires** en moyenne
- Coût des outils professionnels : **30-100€/mois**

Conséquences :

- Productivité limitée (2-3 articles/semaine maximum)
- Fatigue et démotivation
- Coûts élevés pour freelances et étudiants

Objectifs

Objectifs fonctionnels

- Réduire de **80%** le temps entre interview et article finalisé
- Permettre la **personnalisation** de la génération (longueur, ton, structure)
- Garantir une **qualité éditoriale** exploitable

Objectifs pédagogiques

- Maîtriser **FlutterFlow** pour le développement mobile rapide
- Intégrer des **APIs modernes** (IA et transcription)
- Implémenter des **custom functions/actions** en Dart
- Respecter les **contraintes de temps** (10h) et budget (0€)
- Produire un **MVP fonctionnel** démontrable

Objectifs techniques

- Appels API directs (pas de backend complexe)
- **UX fluide** minimaliste

Périmètre et limites

INCLUS dans la v1

Authentification

- Inscription et connexion par email/mot de passe
- Réinitialisation mot de passe par email

Workflow audio → article

- Sélection audio depuis l'appareil (file picker)
- Validation format (MP3) et taille (<25MB)
- Transcription automatique via API IA transcription (groq Whisper)
- Configuration paramètres article (longueur, ton, structure)
- Génération article IA API

Édition et export

- Éditeur de texte
- Compteur caractères et mots en temps réel
- Export '.md' sur web
- Partage via menu système (Copie, email, Drive, WhatsApp...) sur mobile

UX et sécurité

- Alert avant fermeture si article non exporté
- Gestion erreurs explicite (quota, format, réseau)

Personnalisation

- Sauvegarde avec bibliothèque d'articles
- Modification du profil
- Choix des paramètres par défaut
- Suppression du compte / de l'ensemble des articles
- Export de l'ensemble des données utilisateur

EXCLU de la v1

Stockage et historique

- Bibliothèque d'audios (pas de stockage des fichiers possible sur les plans gratuit)
- Versioning des articles pas automatique possible manuellement

Fonctionnalités avancées

- Autocomplétion IA type Copilot dans l'éditeur
- Enregistrement audio in-app (micro intégré)
- Édition audio (trim, noise reduction)
- Export PDF/Word avancé
- Collaboration multi-utilisateurs
- Notifications push
- Mode offline (nécessite connexion internet)
- Multi-langue (UI uniquement en français)

Justification des exclusions :

- **Temps limité** : 10h ne permettent pas ces fonctionnalités
- **Budget 0€** : Firebase Storage et Firestore ont des limites strictes en free tier
- **Simplicité** : Concentration sur les fonctionnalités essentielles

PALETTE DE COULEURS

Couleur	Hex	Usage
DEEP OCEAN	#1A2B4A	Titres bold
ELECTRIC CYAN	#00D9FF	Dégradé : liens et boutons
SUNSET CORAL	#FF6B6B	Fond dégradé : Alertes, accents
GHOST WHITE	#F8F9FD	Arrière-plan
CHARCOAL	#2D3748	Texte principal
SLATE GRAY	#64748B	Texte secondaire

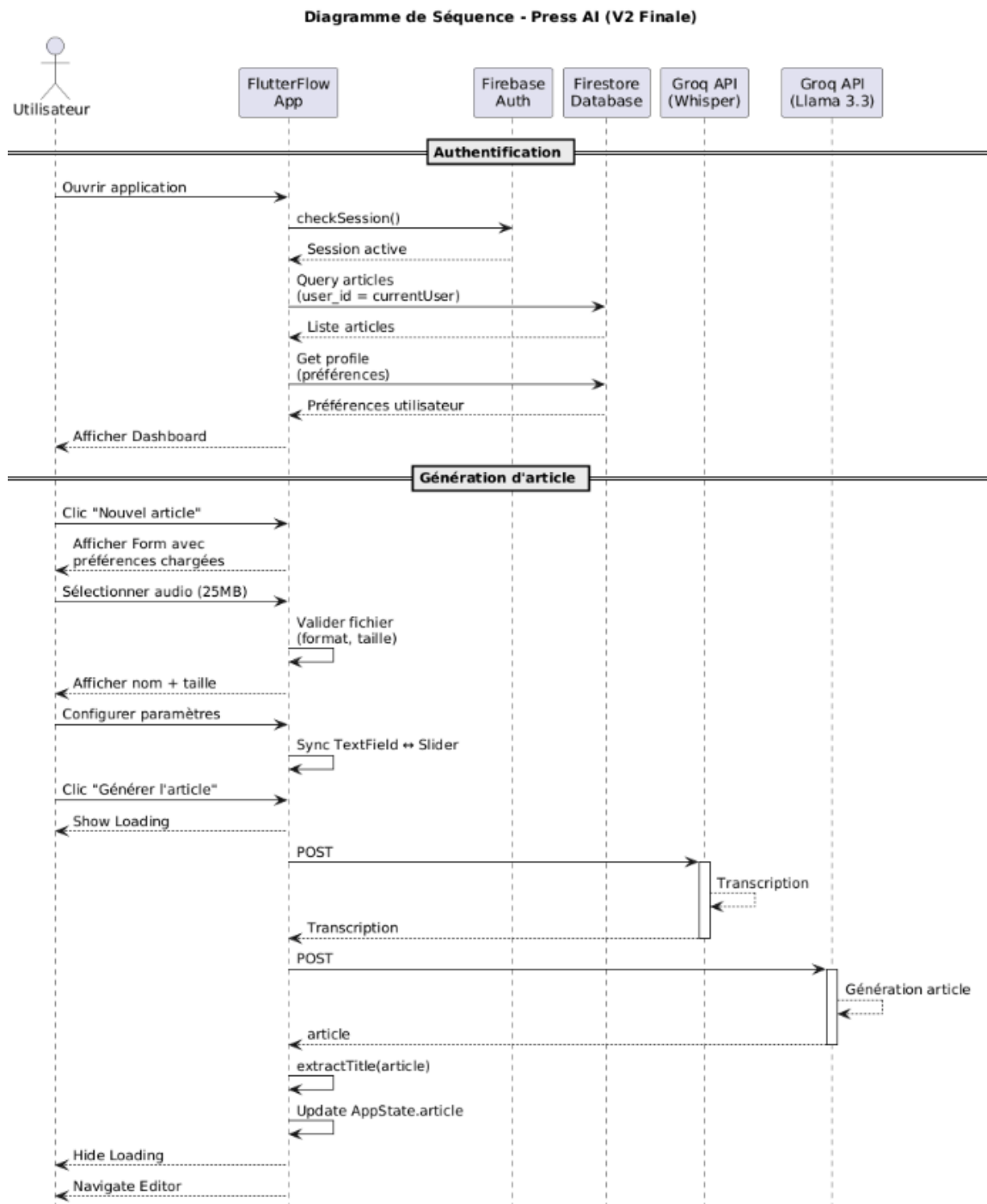
Sémantiques

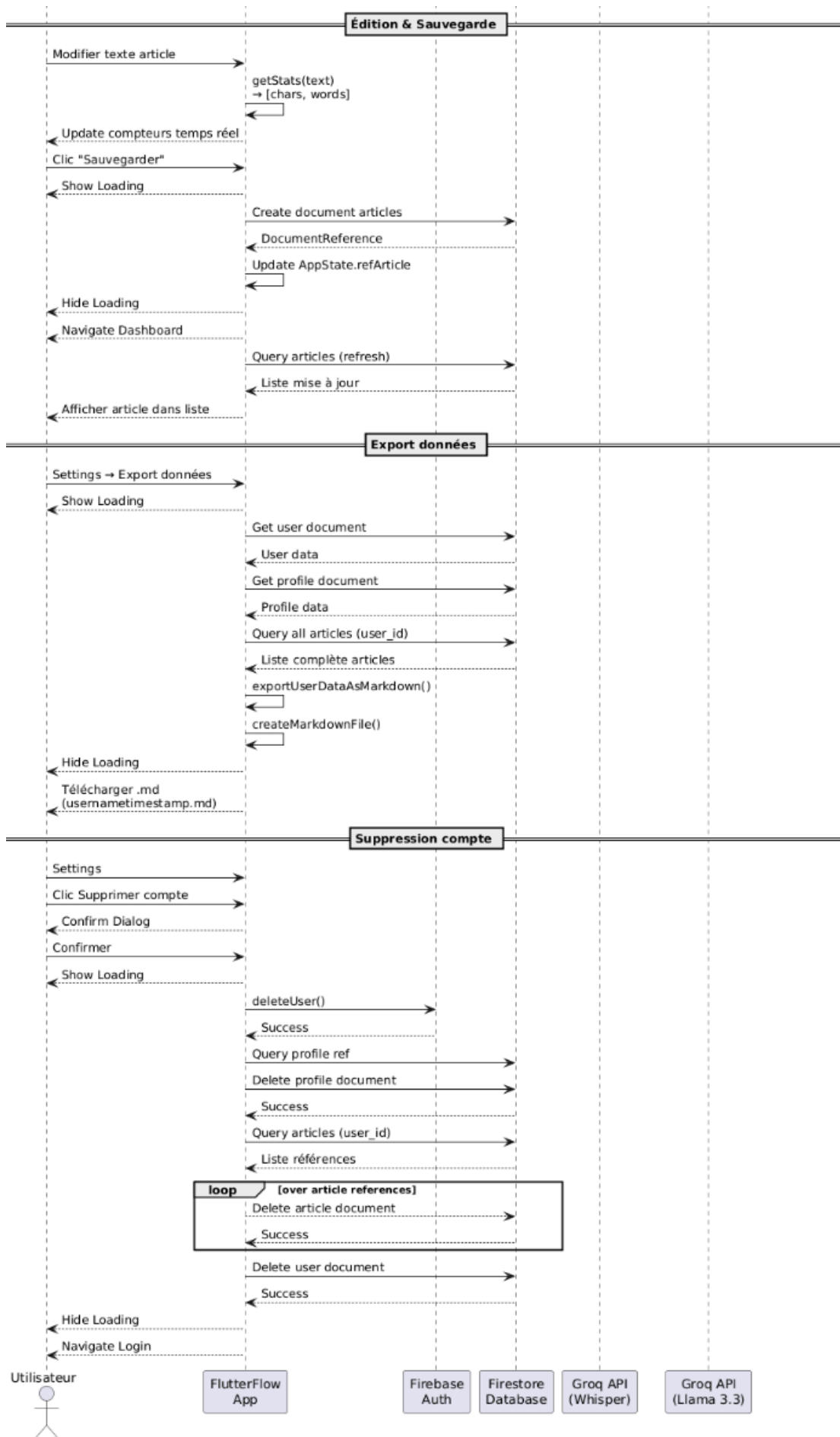
- **Success** : #10B981 • **Warning** : #F59E0B • **Error** : #EF4444

Dégradés

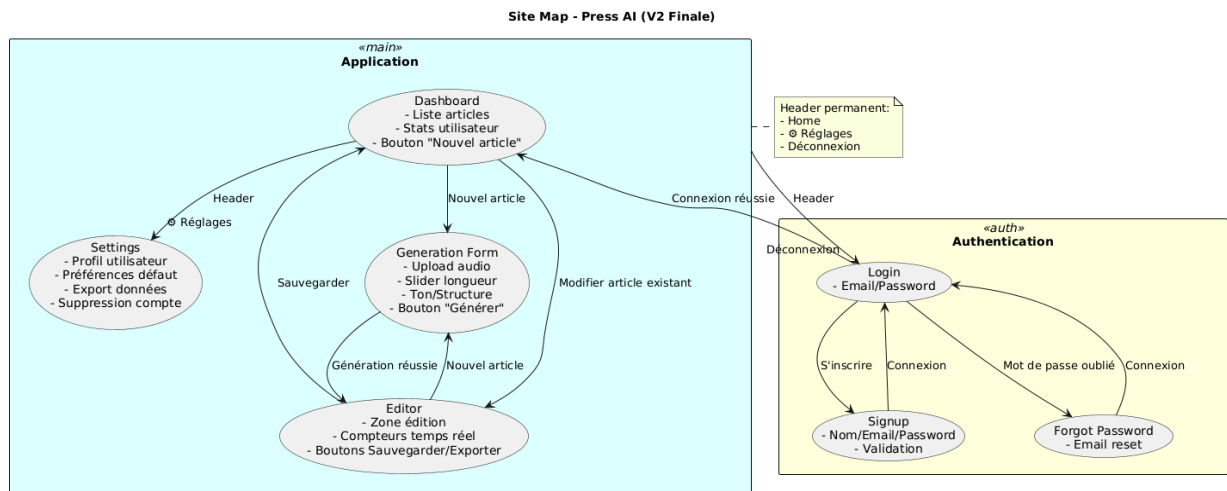
- `background: linear-gradient(135deg, #00D9FF 0%, #0099FF 100%);`
- `background: linear-gradient(90deg, #FF6B6B 0%, #FFB347 100%);`

Exemple d'utilisation par un utilisateur



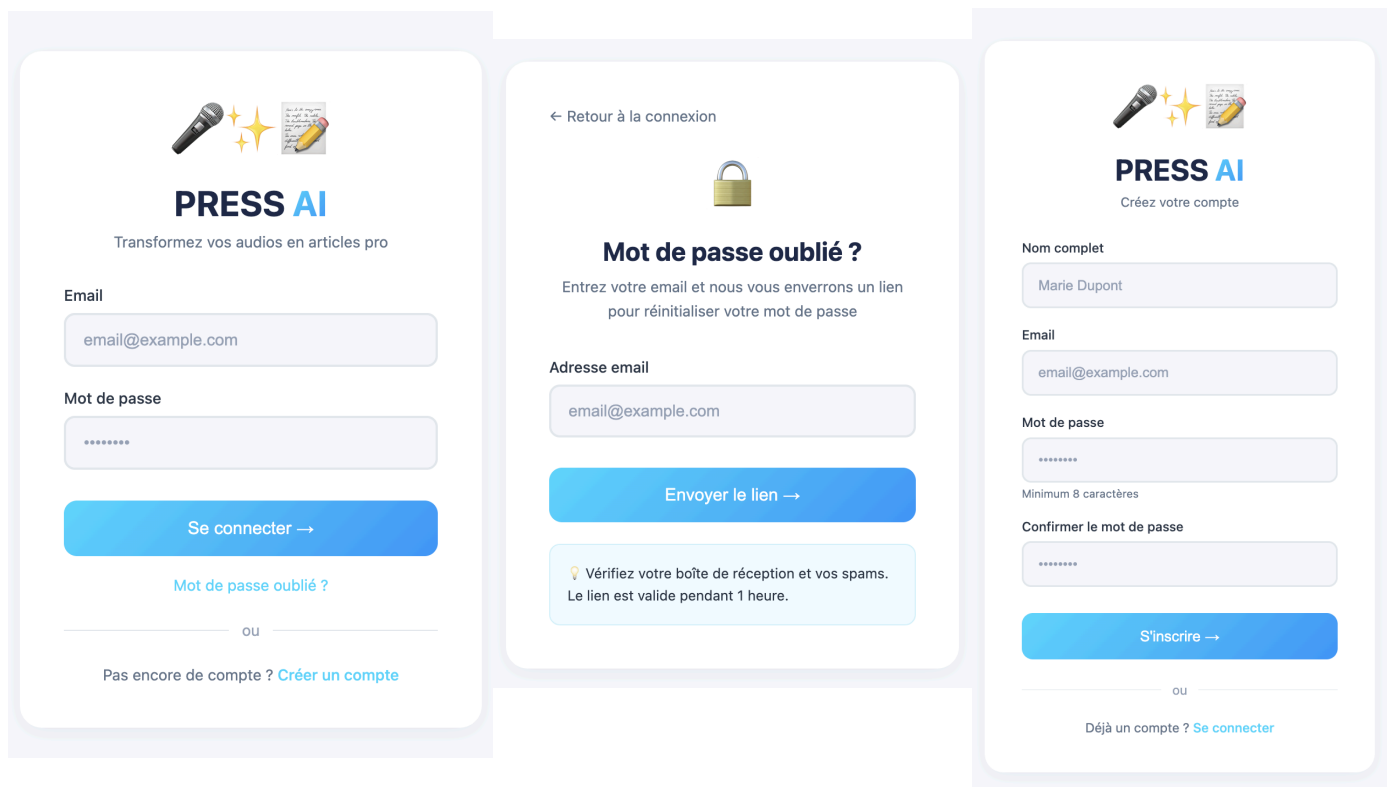


Map des interfaces du site



Wireframe des interfaces

Pages de connexion et d'inscription



Dashboard Utilisateur et Réglages

⚠ Aucune sauvegarde automatique - Exportez vos articles !

PRESS AI
Marie Dupont Déconnexion

Mes articles

5 articles générés + Nouvel article

#1

interview_maire_budget2025.mp3

Le maire de Séné présente le budget 2025 axé sur la transition écologique

28 déc. 2024, 14h30

Le maire de Séné a présenté hier les grandes lignes du budget municipal pour l'année 2025, marqué par des investissements importants dans les infrastructures et l'écologie...

2000 car Neutre Classique

Voir

#2

exposition_artiste_vannes.mp3

Sylvia Martín prépare son exposition à Grain de Sel pour 2027

27 déc. 2024, 16h15

Artiste espagnole installée en Bretagne, Sylvia Martín prépare une exposition à Grain de Sel. Mère d'une fille de 4 mois, elle mêle aquarelle minutieuse et esquisses rapides...

1500 car Dynamique Interview

Voir

#3

conseil_municipal_urbanisme.m4a

Nouveau plan d'urbanisme : trois zones d'aménagement validées

26 déc. 2024, 10h45

#4

festival_jazz_juillet.wav

Festival de jazz : trois jours de concerts en juillet à Arradon

25 déc. 2024, 18h20

Mon compte

Profil

MD Marie Dupont
marie.dupont@presseai.com

Modifier le profil

Mes statistiques

12
Articles

24k
Mots écrits

45
Jours

Préférences

Notifications par email

Longueur par défaut
2000 caractères ▼

Ton éditorial par défaut
Neutre ▼

Structure par défaut
Classique ▼

Mes données

Exportez toutes mes données

Supprimer tous mes articles

Gestion du compte

Changer mon mot de passe

Supprimer mon compte

Formulaire

⚠ Aucune sauvegarde automatique - Exportez vos articles !

PRESS AI
Marie Dupont Déconnexion

Nouveau projet

Transformez votre interview en article professionnel

1. Sélection de l'audio

Glissez votre fichier audio ici
MP3, M4A, WAV - Max 100 MB

Parcourir les fichiers

2. Configuration de l'article

Longueur cible

●

2000 car

Ton éditorial

Neutre

Dynamique

Analytique

Structure

Classique

Q&A

Reportage

Générer l'article

Editeur (faux article d'exemple)

⚠ Aucune sauvegarde automatique - Exportez votre article !

PRESS AI📄 1,847 caractères • 267 mots

B / H2 H3 • Liste

Interview du Maire - Budget 2025

Le maire de Séné a présenté hier les grandes lignes du budget municipal pour l'année 2025, marqué par des investissements importants dans les infrastructures et l'écologie.

Principales annonces

L'édile a annoncé plusieurs mesures phares pour l'année à venir. "Nous allons investir massivement dans la transition écologique de notre commune", a-t-il déclaré lors de la conférence de presse.

Le budget prévoit notamment la rénovation énergétique de trois écoles, l'installation de panneaux solaires sur les bâtiments municipaux, et la création de nouvelles pistes cyclables.

Réactions

L'opposition municipale a salué ces orientations tout en demandant plus de précisions sur le calendrier de mise en œuvre...

Nouvel article Régénérer 🔄 Exporter 📄

SPÉCIFICATION TECHNIQUES

1. ARCHITECTURE DE DONNÉES

1.1 Base de données - Firebase Firestore

Collection **users**

- Stockage des informations d'authentification principales
- Champs: **user_id**, **email**, **user_name**, **created_time**

Collection **profiles**

- Stockage des préférences utilisateur pour la génération d'articles
- Champs:
 - **user** (user référence) : La référence de l'utilisateur associé au profil
 - **length** (int) : Longueur par défaut (500-3000 caractères)
 - **tone** (string) : Ton par défaut (Neutre/Dynamique/Analytique)
 - **structure** (string) : Structure par défaut (Classique/Q&A/Reportage)

Collection **articles**

- Stockage des articles générés par utilisateur
- Champs:
 - **user_id** (string) : Référence utilisateur
 - **title** (string) : Titre extrait automatiquement
 - **body** (string) : Corps complet de l'article en Markdown
 - **target_length** (int) : Longueur cible configurée
 - **created_at** (timestamp) : Date de création

Note importante: Les fichiers audio ne sont pas stockés dans la base de données. Ils sont transmis directement aux APIs puis supprimés après traitement pour optimiser les coûts de stockage.

2. INTÉGRATIONS API EXTERNES

2.1 Groq API - Transcription audio (Whisper Large v3)

Endpoint: <https://api.groq.com/openai/v1/audio/transcriptions>

Modèle: [whisper-large-v3](#)

Utilisation: Conversion de fichiers audio en texte français

Contraintes:

- Limite fichier: 25 MB maximum
- Formats acceptés: MP3, M4A, WAV
- Gratuit avec quotas généreux

L'API builder natif FlutterFlow ne permet pas la transmission de fichiers audio (type [FFUploadedFile](#)) comme paramètre de requête. Une Custom Action Dart sera développée pour gérer l'upload multipart/form-data et la conversion des bytes audio (custom action : transcribeAudio).

2.2 Groq API - Génération d'articles (Llama 3.3 70B)

Endpoint: <https://api.groq.com/openai/v1/chat/completions>

Modèle: [llama-3.3-70b-versatile](#)

Utilisation: Transformation de transcriptions en articles journalistiques formatés

Paramètres configurables:

- [target_length](#): Longueur cible en caractères
- [tone](#): Style éditorial (neutre/dynamique/analytique)
- [structure](#): Format de l'article (classique/interview/reportage)
- [temperature](#): 0 (pas de créativité pur un article on n'invente pas de faits)
- [max_tokens](#): 4000

Sortie: Article complet en format Markdown avec titre et corps structuré

Avec le premier appel API fait de cette manière c'était plus pratique de tout faire en JSON avec des custom action.

3. CUSTOM ACTIONS & CUSTOM FUNCTIONS

3.1 Custom Actions

transcribeAudioGroq

Paramètres:

- `audioFile` (FFUploadedFile) : Fichier audio sélectionné

Return: `Map<String, dynamic>` avec clés `success` (bool), `error` (String), `data` (String), `details` (String).

Description: Envoie le fichier audio à Groq Whisper via requête multipart/form-data, récupère la transcription et gère les erreurs (fichier invalide, quota dépassé, erreur réseau).

generateArticleGroq

Paramètres:

- `transcription` (String) : Texte transcrit
- `targetLength` (int) : Longueur cible
- `tone` (String) : Ton éditorial
- `structure` (String) : Structure article

Return: `Map<String, dynamic>` avec clés `success` (bool), `article` (String)

Description: Construit un prompt système et utilisateur optimisé, appelle Groq Llama, parse la réponse et retourne l'article formaté en Markdown. S'il y a une erreur elle est stockée dans "article" et apparaît à la place.

exportUserDataAsMarkdown

Paramètres:

- `userId` (String) : Identifiant utilisateur

Return: `String` (contenu Markdown complet)

Description: Récupère toutes les données utilisateur depuis Firestore (profil, articles), compile les statistiques, formate en document Markdown structuré avec métadonnées pour chaque article.

createMarkdownFile

Paramètres:

- `userName` (String) : nom utilisateur
- `text` (String) : contenu Markdown

Return: `FFUploadedFile` (fichier Markdown "user_name-timestamp.md")

Description: Construit un fichier Markdown avec la string fournie. Le fichier est nommé avec le nom de l'utilisateur et le timestamp. Cette action est utile pour exporter soit l'ensemble des données utilisateur soit un article.

3.2 Custom Functions

`smartDateDisplay`

Paramètres: `createdAt` (DateTime)

Return: `String`

Description: Affiche "Il y a X minutes/heures" si < 24h, "Hier" si < 48h, puis date complète. Optimise la lisibilité des timestamps.

`getTitle`

Paramètres: `article` (String)

Return: `String`

Description: Parse le Markdown pour extraire le titre (première ligne commençant par #).

`countAllWord`

Paramètres: `articles` (List<ArticlesRecord>)

Return: `int`

Description: Parcourt tous les articles, nettoie le Markdown (supprime #, **, *), compte les mots totaux pour statistiques utilisateur.

`validateLength`

Paramètres: `input` (String)

Return: `int`

Description: Parse et valide l'entrée utilisateur pour longueur d'article, entre 500 et 3000, retourne valeur par défaut (1000) si erreur.

getStats**Paramètres:** `text` (String)**Return:** `List<int>` [caractères, mots]**Description:** Fonction combinée retournant simultanément le nombre de caractères et de mots d'un texte. Nettoie la syntaxe Markdown (supprime #, **, *, liens) avant comptage des mots. Utilisée pour affichage temps réel des statistiques dans l'éditeur. Index 0 = caractères bruts, Index 1 = mots nettoyés.

FlutterFlow impose des types stricts dans l'interface visuelle (Text widgets, Conditional actions) et refuse les accès directs comme `json['key']` sur type `dynamic`. Ces fonctions wrapper permettent de manipuler les réponses Custom Actions des API.

isOK**Paramètres:** `json` (dynamic)**Return:** `bool`**Description:** Parse le champ `success` d'une réponse JSON typée `dynamic`. Utilisée pour évaluer le succès des Custom Actions retournant des Maps (transcription, génération, export).**getStringFromJson****Paramètres:**

- `json` (dynamic) : Objet JSON
- `param` (String) : Clé à extraire

Return: `String`**Description:** Extrait et convertit en String la valeur d'une clé spécifique dans un objet JSON typé `dynamic`. Permet d'accéder aux champs des réponses API sans erreur de typage.

4. SYSTÈME D'AUTHENTIFICATION

Méthode: Email et mot de passe uniquement

Workflow inscription:

1. Validation côté client (email valide, mot de passe \geq 6 caractères)
2. Création compte Firebase Auth
3. Création document `users` avec `user_name`, `email`, `created_time`
4. Création document `profiles` avec préférences par défaut
5. Redirection Login

Workflow connexion:

1. Firebase Auth vérifie identifiants
2. Récupération session utilisateur
3. Chargement articles depuis Firestore
4. Chargement du profil depuis Firestore
5. Redirection Dashboard

Récupération mot de passe:

- Utilise `sendPasswordResetEmail()` Firebase Auth
- Envoi automatique email avec lien temporaire
- Interface dédiée "Mot de passe oublié" avec un input mail

Modification mot de passe:

- Accessible depuis page Réglages
- Confirmation utilisateur (dialog)
- Utilise `updatePassword()` Firebase Auth
- Nécessite ré-authentification récente

Suppression compte:

- Accessible depuis page Réglages
- Confirmation utilisateur (dialog)
- Suppression tous articles (`deleteAllArticles`)
- Suppression document `profiles`
- Suppression document `users`
- Suppression compte Firebase Auth
- Redirection page connexion

Champs utilisateur:

- `email` (string, unique, requis)
- `password` (hashé, géré par Firebase)
- `user_name` (string, modifiable)

5. PAGES & NAVIGATION




Pages publiques (non authentifiées):

1. **Login** : Connexion email/password
2. **Signup** : Inscription nouveau compte
3. **Forgot Password** : Réinitialisation mot de passe

Pages protégées (authentification requise):

4. **Dashboard** : Bibliothèque articles, statistiques utilisateur
5. **Generation Form** : Configuration et lancement génération article
6. **Editor** : Édition, visualisation, export article
7. **Settings (Réglages)** : Gestion profil, préférences, données

Header permanent (sur toutes les pages authentifiées):

- Logo "PRESS AI" : Navigation vers Dashboard
- Bouton " Accueil" : Retour Dashboard
- Bouton " Réglages" : Ouvre page Settings
- Info utilisateur " [Nom]"
- Bouton "Déconnexion"

6. GESTION D'ÉTAT - AppState Variables

Synchronisation formulaire:

- `targetLength` (int, défaut: 1000) : Coordonne TextField et Slider longueur article

Compteurs éditeur temps réel:

- `wordCount` (int) : Nombre de mots article en cours
- `charCount` (int) : Nombre de caractères article en cours

Gestion erreurs:

- `error` (String, nullable) : Message d'erreur global

Transit données inter-pages:

- `article` (String) : Contenu article généré/édité, transit Form → Editor → Dashboard
- `refArticle` (DocumentReference, nullable) : Référence Firestore si article sauvegardé, null sinon

Liste articles utilisateur:

- `articlesList` (List<DocumentReference>) : Références Firestore tous articles, chargé au login et refresh Dashboard

7. CONTRAINTES TECHNIQUES

7.1 Plateformes de déploiement

Plateforme principale: Web (navigateur)

- Déployé sur hébergement FlutterFlow gratuit
- URL publique accessible
- Compatible tous navigateurs modernes (Chrome, Safari, Firefox, Edge)

Plateforme secondaire: Android

- APK téléchargeable générée via FlutterFlow
- Non Testé

iOS: Non déployé

- Contraintes Apple (certificats, compte développeur \$99/an)
- Compensée par accessibilité web universelle

7.2 Responsive Design

Approche: Mobile-first avec adaptation progressive

Breakpoints personnalisés:

Mobile: < 810px	Tablette: 810-1024px	Desktop: > 1024px	Wide: > 1440px
-----------------	----------------------	-------------------	----------------

Adaptations spécifiques:

- Dashboard: Grid 2/3 colonnes selon largeur écran avec 3/7/10/14 lignes
- Editor: Toolbar responsive

Compense l'absence de déploiement mobile natif en garantissant expérience optimale sur tous appareils via navigateur.

7.3 Limitations techniques

Audio:

- Taille maximale: 25 MB (contrainte API Groq Whisper)
- Formats: MP3

Articles:

- Pas de pagination implémentée
- Chargement complet liste (acceptable < 100 articles)
- OrderBy `created_at` DESC (plus récents en premier)

8. SÉCURITÉ & GESTION DES ERREURS

8.1 Sécurité Firestore

Validation données:

- Côté client: Validation formulaires (longueur, formats)
- Côté Firestore: Rules vérifient ownership avant write

8.2 Gestion des erreurs

Firestore Auth:

- Messages erreurs natifs Firestore affichés automatiquement
- Exemples: "Email invalide", "Mot de passe trop court"

Génération d'articles:

- Vérification audio sélectionné → Message dans Form
- Erreur transcription → Message dans Form
- Erreur génération → Message dans Editeur

9. TESTS & VALIDATION

9.1 Méthodologie de test

Type: Tests manuels exploratoires

Avec: 3 comptes utilisateurs distincts

Navigateurs/Appareils:

- Chrome (Windows, macOS)
- Safari (macOS, iOS)
- Appareils: PC Desktop, iPhone, Smartphone Android

GESTION DE PROJET

Diagramme de Gantt

