

DOCUMENTATION DU PROJET HYPERMEDIA

Analyse texte brut & nuage de mots

Rédigé par :

Litissia LARBI

SOMMAIRE

1.	Présentation du Projet.....	2
1.1	Objectifs	2
2.	Flux de Traitement.....	2
2.1	Acquisition des Données	2
2.2	Normalisation du Texte.....	2
2.3	Tokenisation et Filtrage.....	2
2.4	Analyse de Fréquence	2
2.5	Visualisation.....	3
2.6	Export.....	3
3.	Diagramme d'Architecture	3
4.	Technologies Utilisées	4
4.1	Technologies Front-End	4
4.2	Ressources Externes	4
4.3	APIs Web Utilisées	4
4.4	Aperçu de l'interface réalisée.....	4

1. PRESENTATION DU PROJET

NuageMaker est une application web interactive permettant de générer des nuages de mots à partir de textes fournis par l'utilisateur. L'application analyse automatiquement le texte, effectue un traitement linguistique en français, et produit des visualisations esthétiques et informatives de mots clés.

1.1 Objectifs

- Fournir une visualisation intuitive de la fréquence des mots dans un texte
- Appliquer un traitement linguistique intelligent (tokenisation, normalisation)
- Offrir une interface utilisateur moderne et réactive
- Permettre l'exportation des visualisations générées

2. FLUX DE TRAITEMENT

Le traitement du texte s'effectue en plusieurs étapes successives garantissant la qualité de l'analyse linguistique :

2.1 Acquisition des Données

- **Saisie manuelle** : L'utilisateur saisit ou colle le texte dans la zone de texte dédiée
- **Importation de fichier** : Chargement d'un fichier .txt avec encodage UTF-8 via FileReader API.

2.2 Normalisation du Texte

Cette phase prépare le texte pour l'analyse en appliquant plusieurs transformations :

- **Conversion en minuscules** : Uniformisation de la casse pour éviter les doublons
- **Normalisation des apostrophes** : Remplacement des variantes typographiques (', ') par l'apostrophe standard
- **Suppression de la ponctuation** : Retrait des caractères spéciaux et symboles (.,:;!?-«»...)
- **Conservation des caractères français** : Préservation des accents et caractères spéciaux (é, è, à, ç, œ)
- **Consolidation des espaces** : Réduction des espaces multiples en un seul espace

2.3 Tokenisation et Filtrage

- **Découpage en tokens** : Séparation du texte en mots individuels via split() sur les espaces.
- **Filtrage par longueur** : Élimination des mots de moins de 3 caractères.
- **Suppression des mots vides** : Retrait des mots grammaticaux sans valeur sémantique (le, la, de, et, un...) à partir d'un fichier de mots vides .

2.4 Analyse de Fréquence

- **Comptage des occurrences** : Calcul de la fréquence de chaque mot unique
- **Tri par fréquence** : Classement décroissant des mots selon leur nombre d'apparitions

- **Limitation à 60 mots** : Sélection des mots les plus fréquents pour optimiser la lisibilité

2.5 Visualisation

- **Génération du nuage** : Utilisation de wordcloud2.js avec configuration personnalisée (grille 18px, rotations limitées, palette de 15 couleurs)
- **Liste des mots-clés** : Affichage structuré avec barres de progression proportionnelles
- **Statistiques** : Présentation du nombre de mots traités, mots uniques et mots supprimés

2.6 Export

Conversion du canvas en image PNG via toBlob() et téléchargement automatique avec nom horodaté.

3. DIAGRAMME D'ARCHITECTURE

Architecture 100% Client-Side : Utilisation exclusive de HTML5, CSS3 et JavaScript pour une exécution instantanée sans serveur. L'application suit une architecture client-side en trois couches (Figure 1) :

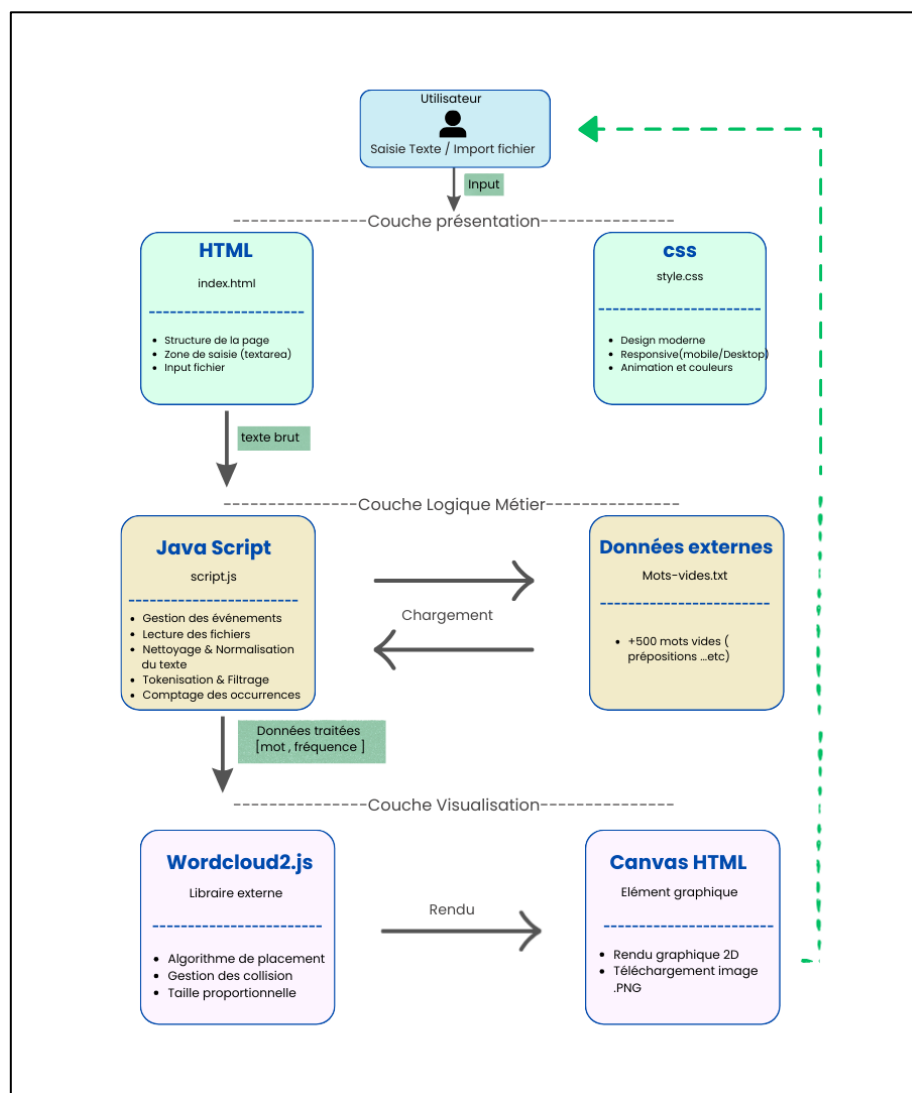


Figure 1 : Le diagramme d'architecture

4. TECHNOLOGIES UTILISEES

4.1 Technologies Front-End

Tableau 1 : Organisation des technologies Front-End

Technologie	Version	Utilisation
HTML5	-	Structure sémantique, Canvas, FileReader API
CSS3	-	Design responsive, gradients, transitions, flexbox
JavaScript	ES6+	Logique métier, manipulation DOM, événements
Wordcloud2.js	1.2.2	Génération et rendu des nuages de mots

4.2 Ressources Externes

- **CDN cloudflare** : Hébergement de la bibliothèque wordcloud2.js
- **Fichier mots-vides.txt** : Liste de 500+ mots vides français pour le filtrage linguistique

4.3 APIs Web Utilisées

Tableau 2 : Les different APIs Web utilisees

API	Fonction
FileReader API	Lecture asynchrone des fichiers texte téléversés
Canvas 2D Context	Dessin vectoriel du nuage de mots et export PNG
Fetch API	Chargement asynchrone du fichier mots-vides.txt
Blob API	Conversion du canvas en fichier image téléchargeable

4.4 Aperçu de l'interface réalisée

