



Projet final

# Table des matières

A. Contexte	p. 1
B. Fonctionnalités	p. 1
C. Focus sur les fichiers Markdown	p. 2
i. Organisation	p. 2
ii. Rédaction de contenu	p. 2
iii. Prévisualisation	p. 2
iv. Import et export	p. 2
D. Focus sur les blocs personnalisés	p. 3
i. Construction des blocs	p. 3
ii. Raccourcis claviers	p. 3
iii. Bibliothèque de blocs	p. 3
iv. Import et export	p. 3
E. Focus sur les images	p. 4
i. Bibliothèque	p. 4
ii. Import et export	p. 4
F. Contraintes supplémentaires	p. 5
i. Techniques	p. 5
ii. Graphique	p. 5
iii. Stockage	p. 5
G. Fonctionnalités bonus	p. 5
H. Modalités	p. 6
i. Le rendu du projet	p. 6
ii. Bonus potentiels	p. 6
iii. Malus potentiels	p. 6

## Contexte

Votre entreprise souhaite créer une application pour gérer la rédaction de documents interne et vous a missionné sur sa création.

Cette application doit permettre de gérer des fichiers au format Markdown, de les afficher dans un navigateur au format HTML, de créer des blocs HTML/Markdown personnalisés et dispose d'un système de bibliothèque pour la gestion des images.

## Fonctionnalités

L'application offrira les fonctionnalités suivantes :

- Fichiers Markdown :
  - Organisation dans une arborescence (dans des dossiers)
  - Gestion des fichiers Markdown
  - Rédaction du contenu Markdown
  - Prévisualisation
  - Import et export des fichiers
- Blocs personnalisés :
  - Création de blocs personnalisés en HTML/Markdown
  - Gestion des blocs personnalisés sous forme d'une bibliothèque de blocs
  - Affectation de raccourcis claviers pour insérer les blocs personnalisés dans un fichier Markdown
  - Import et export des blocs personnalisés
- Bibliothèque d'images :
  - Import d'images via bouton ou glisser-déposer dans la bibliothèque
  - Gestion des images dans la bibliothèque
  - Insertion d'images de la bibliothèque dans un fichier Markdown
  - Import et export des images

# Focus sur les fichiers Markdown

## Organisation

Les fichiers Markdown seront organisés sous forme d'une arborescence, c'est-à-dire de dossiers pouvant contenir des fichiers et/ou d'autres dossiers.

L'interface listant les fichiers doit donc prévoir à la fois la gestion des fichiers et des dossiers ; en détails (liste non exhaustive) :

- Création d'un dossier à la racine ou dans un dossier parent
- Renommage d'un dossier
- Suppression d'un dossier
- Déplacement d'un dossier dans un autre dossier ou à la racine en drag'n'drop
- Création d'un nouveau fichier à la racine ou dans un dossier parent
- Renommage de fichier
- Suppression de fichier
- Ouverture d'un fichier pour modification
- Déplacement d'un fichier dans un dossier ou à la racine en drag'n'drop

## Rédaction de contenu

La rédaction du contenu d'un fichier Markdown se fera dans une interface dédiée. Cette interface doit également permettre d'accéder aux fonctionnalités d'insertion d'images ou blocs personnalisés à partir de leurs bibliothèques respectives.

## Prévisualisation

La prévisualisation se fera sous forme d'une interface dédiée affichant le code Markdown converti en HTML.

### Notes

La conversion Markdown <-> HTML n'étant pas l'objectif de ce projet, il est recommandé d'utiliser une bibliothèque de conversion comme Showdown (<http://showdownjs.com>) ou Marked (<https://marked.js.org>).

## Import et export

L'application doit permettre d'exporter un fichier Markdown sous forme d'un fichier `.md`.

L'application doit également permettre d'importer un fichier Markdown en l'injectant dans l'organisation des fichiers.

## Focus sur les blocs personnalisés

### Construction des blocs

L'application doit permettre la rédaction de blocs personnalisés réutilisables. Tout comme le contenu d'un fichier Markdown, les blocs personnalisés peuvent contenir à la fois du HTML et du Markdown.

### Raccourcis claviers

Pour faciliter la rédaction des fichiers Markdown, les blocs personnalisés pourront être associés à des raccourcis claviers uniques, via une interface dédiée.

Lors de la rédaction du contenu d'un fichier Markdown dans l'interface dédiée, l'exécution d'un raccourci clavier insèrera automatiquement le bloc associé.

### Bibliothèque de blocs

L'ensemble des blocs personnalisés créés seront visibles et gérables à partir de la bibliothèque de blocs. S'agissant du point central de gestion, la bibliothèque permettra (liste non exhaustive) :

- La création de blocs personnalisés
- La modification d'un bloc personnalisé
- La suppression d'un bloc personnalisé
- Le renommage d'un bloc personnalisé
- L'import / l'export de blocs personnalisés

### Import et export

L'application doit permettre d'exporter un bloc personnalisé sous forme d'un fichier `.part.md1c` et une liste de blocs personnalisés sous forme d'un fichier unique `.parts.md1c`.

L'application doit également permettre d'importer un bloc ou une liste de blocs, à partir de fichiers aux formats cités ci-dessus.

# Focus sur les images

## Bibliothèque

L'ensemble des images utilisables dans un fichier Markdown sont stockées dans une bibliothèque dédiée.

Les images seront stockées sous forme de base64 (<https://fr.wikipedia.org/wiki/Base64>), afin d'éviter la gestion de binaires.

S'agissant du point central de gestion, la bibliothèque permettra (liste non exhaustive) :

- L'import d'images via bouton "parcourir" ou glisser/déposer (conversion base64)
- La prévisualisation d'images
- Le renommage d'une image
- La suppression d'une image
- L'import / l'export des images par fichiers spécifiques

## Import et export

L'application doit permettre d'exporter une image sous forme d'un fichier `.img.mdl` et une liste d'images sous forme d'un fichier unique `.imgs.mdlc`.

L'application doit également permettre d'importer une image ou une liste d'image à partir de fichiers aux formats cités ci-dessus.

# Contraintes supplémentaires

## Techniques

Le projet devra être pur front et réalisé avec React ; les bibliothèques React-Router et React-Redux devront également être utilisées.

## Graphique

L'application devra être un minimum stylisée ; s'il n'est pas nécessaire de penser une charte graphique complète, un minimum de design sera requis afin de rendre son utilisation agréable.

## Stockage

Les données devront être stockées dans le navigateur de manière (pseudo-)persistant. L'application devra donc utiliser le localStorage ou IndexedDB.

## Fonctionnalités bonus

Le projet peut se développer bien au delà des fonctionnalités listées.

Exemple de fonctionnalités hors notation mais pouvant apporter des points bonus (liste non exhaustive) :

- Système de chiffrement par mot de passe des exports avec déchiffrement à l'import
- Interface 100% responsive
- Thèmes graphiques de l'interface (zoom, couleurs, etc.)
- Système de profils ("perso", "travail", etc.) avec bibliothèques dédiées

## Modalités

Les modalités de ce projet sont les suivantes :

- Ce projet devra être réalisé en groupe de 3 à 4 personnes
- Votre rendu devra être effectué dans une archive ZIP envoyée par NetYparéo, nous vous conseillons donc une taille inférieure à 25Mo (voir livrables).
- Le projet doit être "pur front", son fonctionnement ne doit donc pas reposer sur l'utilisation d'un serveur.
- Mis à part ceux explicitement interdits, vous êtes autorisés à installer des packages, tant qu'ils ne permettent pas de résoudre une fonctionnalité.

## Le rendu du projet

Le projet doit être rendu avant la date limite sur NetYparéo.

Le projet devra inclure un fichier README.md contenant la liste des membres du groupe ainsi que toute information nécessaire à la mise en place du projet.

Attention à la taille de votre rendu ; un rendu trop volumineux, au-delà de 25Mo, pourra être sanctionné.

## Bonus potentiels

L'ajout de fonctionnalités supplémentaires ou un travail particulièrement qualitatif sur un élément du projet peut donner droit à des points supplémentaires.

## Malus potentiels

Des malus pourront être appliqués :

- En cas de copier/coller trop important d'une source, que ce soit internet, un autre groupe ou une IA générative
- En cas de copier/coller non sourcé dans un commentaire
- Retard d'envoi du projet
- Hors sujet