

Table des matières

I.	Introduction.....	2
II.	Objectifs	2
III.	Architecture	2
IV.	Technologies utilisées	5
V.	Installation et configuration	5
VI.	Configuration des variables d'environnement	5
VII.	Fonctionnalités.....	6
VIII.	API Endpoints	6
IX.	Authentification et Sécurité	6
X.	Utilisation.....	6
XI.	Conclusion.....	8

Documentation du Projet 4FAP

I. Introduction

HostIT est une plateforme de stockage évolutive conçue pour les utilisateurs non techniques. Elle permet aux utilisateurs de télécharger, gérer et partager des fichiers en toute sécurité. Le projet est divisé en trois principales parties : le backend, le frontend et l'application de bureau.

II. Objectifs

- Fournir une interface utilisateur intuitive pour le téléchargement et la gestion des fichiers.
- Assurer une sécurité robuste avec des mécanismes d'authentification et d'autorisation.
- Offrir une solution de stockage évolutive avec intégration AWS S3.

III. Architecture

L'architecture du projet HostIT est basée sur une approche client-serveur. Elle comprend les composants suivants :

1. **Backend (Node.js + Express)** : Gère les requêtes HTTP, l'authentification, l'autorisation et les interactions avec la base de données MongoDB.
2. **Frontend (React.js)** : Fournit une interface utilisateur dynamique et réactive.
3. **Application de bureau (Electron)** : Offre une solution de bureau pour les utilisateurs qui préfèrent une application native.



IV. Technologies utilisées

1. **Backend** : Node.js, Express.js, MongoDB, Mongoose, JWT pour l'authentification.
2. **Frontend** : React.js, Redux pour la gestion de l'état, Material-UI pour les composants UI.
3. **Application de bureau** : Electron.js.
4. **Stockage** : AWS S3 pour le stockage des fichiers.

V. Installation et configuration

Prérequis :

- A. Node.js
- B. MongoDB
- C. Compte AWS (pour S3)

VI. Configuration des variables d'environnement

Créez un fichier **.env** dans le répertoire backend avec les informations suivantes :

- MONGO_URI=mongodb://localhost:27017/hostit
- JWT_SECRET=your_jwt_secret
- AWS_ACCESS_KEY=your_aws_access_key
- AWS_SECRET_KEY=your_aws_secret_key
- AWS_REGION=your_aws_region
- S3_BUCKET_NAME=your_s3_bucket_name

VII. Fonctionnalités

- **Authentification et autorisation** : Inscription, connexion, déconnexion.
- **Gestion des fichiers** : Téléchargement, suppression, visualisation des fichiers.
- **Quota d'utilisation** : Surveillance de l'espace de stockage utilisé par utilisateur.
- **Application de bureau** : Téléchargement et gestion des fichiers via une application native.

VIII. API Endpoints

1. **Authentification** :
 - POST /api/auth/register : Inscription d'un nouvel utilisateur.
 - POST /api/auth/login : Connexion d'un utilisateur.
2. **Fichiers** :
 - POST /api/files/upload : Téléchargement d'un fichier.
 - DELETE /api/files/delete : Suppression d'un fichier.
 - GET /api/users/quota : Obtention du quota d'utilisation.

IX. Authentification et Sécurité

- Utilisation de JWT (JSON Web Tokens) pour l'authentification et l'autorisation.
- Les mots de passe des utilisateurs sont hachés avec bcrypt avant d'être stockés dans la base de données.
- Les fichiers sont stockés de manière sécurisée sur AWS S3.

X. Utilisation

Interface Web

1. Accédez à l'application via <http://localhost:3000>.
2. Inscrivez-vous ou connectez-vous pour accéder au tableau de bord.

Application de bureau

1. Installez l'application empaquetée.
2. Ouvrez l'application et connectez-vous avec vos identifiants.

XI. Conclusion

HostIT est une plateforme complète de gestion de fichiers conçue pour les utilisateurs non techniques. Grâce à son interface utilisateur intuitive et ses fonctionnalités robustes, elle permet une gestion facile et sécurisée des fichiers. L'intégration avec AWS S3 assure une évolutivité et une fiabilité optimales.