



Application Angular de Gestion Avancée de Documents avec Authentification et Contrôle d'Accès

Réalisé par:

ABOUMOUSSA Ghita

lien de la video :

<https://www.youtube.com/watch?v=Ak0Ic8kHmKw&feature=youtu.be>

Introduction Générale :

Avec l'avènement de la technologie et la numérisation croissante de l'information, la gestion de documents est devenue un enjeu majeur pour de nombreuses organisations. La nécessité de stocker et d'organiser efficacement des documents est essentielle pour garantir une collaboration fluide et une gestion de l'information sans heurts.

Dans ce contexte, le projet Angular que nous proposons vise à développer une application de gestion de documents utilisateur, offrant des fonctionnalités avancées pour la gestion des utilisateurs et des documents.

L'objectif principal de cette application est de répondre aux besoins de deux types d'utilisateurs : les administrateurs et les utilisateurs standards. Les administrateurs sont chargés de la gestion des comptes utilisateur, tandis que les utilisateurs standards sont responsables de la gestion des documents et des dossiers. L'application, construite en utilisant le framework Angular, doit permettre une gestion efficace des informations associées aux documents et aux dossiers.

Ce rapport détaillé mettra en lumière les différentes facettes du projet, de sa conception à sa mise en œuvre, en passant par les choix architecturaux et les fonctionnalités clés. En se concentrant sur des aspects cruciaux tels que la gestion des utilisateurs, ainsi que la manipulation des documents et des dossiers, il fournira une vue d'ensemble complète du potentiel de l'application.

Table de matière

Chapitre 1: Introduction du projet:	4
1. Contexte du projet:	5
2. Problématique:	6
3. Objectifs:	6
4. Conclusion:	7
Chapitre 2: Modélisation	8
1. Diagramme de cas d'utilisation:	9
2. Diagramme d'activités:	12
2. Conclusion:	13
Chapitre 3 : Outils et Technologies :	14
1. Front-End :	15
2. Back-End :	16
3. Développement et Modélisation:	17
4. Conclusion:	17
Chapitre 4 : Réalisation	19
1. Authentification:	20
a) Connexion:	20
b) Inscription:	21
2. Interface Administrateur :	23
3. Interface utilisateur standard :	24
a) Liste des dossiers :	24
b) Ajouter nouveau dossier :	24
c) Liste des documents :	25
d) Ajouter un nouveau document :	26
4. Déconnexion :	26
5. Conclusion :	26
Chapitre : Conclusion :	27

Chapitre 1: Introduction du projet:

Dans ce chapitre, nous amorcerons notre exploration du projet en contextualisant son domaine d'étude et en exposant la problématique à laquelle il se propose de répondre. De plus, nous définirons clairement les objectifs fondamentaux qui guideront notre travail, jetant ainsi les bases essentielles pour une compréhension approfondie de l'ensemble de notre démarche à venir.

1. Contexte du projet:

Avec la montée en puissance de l'informatique dans notre société, la gestion efficace de documents et de données utilisateur est devenue un défi majeur pour de nombreuses organisations et entreprises. Ces documents peuvent varier en nature, depuis des rapports d'entreprise et des fichiers médicaux jusqu'aux simples documents de texte. La nécessité de stocker, organiser, et sécuriser ces données est cruciale pour garantir la continuité des opérations et la conformité aux réglementations en vigueur.

Cependant, la gestion de documents va bien au-delà de la simple sauvegarde de fichiers. Elle englobe également la gestion des autorisations d'accès, la traçabilité des modifications, la création de hiérarchies de dossiers, et bien plus encore. L'un des principaux enjeux est de permettre à différents utilisateurs d'interagir avec ces documents tout en maintenant une sécurité adéquate et des rôles d'accès appropriés.

Dans ce contexte, notre projet vise à développer une application Angular dédiée à la gestion de documents utilisateur, avec une attention particulière portée à l'authentification et à l'autorisation. Nous souhaitons simplifier la gestion des informations associées aux documents et aux dossiers, tout en offrant une expérience utilisateur fluide et sécurisée. Cette application devrait permettre aux administrateurs de gérer les utilisateurs et aux utilisateurs standard de gérer des documents et des dossiers, avec une interface conviviale pour toutes les parties prenantes. Pour atteindre cet objectif, nous devons relever divers défis, allant de la gestion des utilisateurs et des documents à l'implémentation de fonctionnalités de sécurité robustes.

2. Problématique:

La problématique centrale de ce projet réside dans la nécessité de concevoir une application robuste qui permette aux utilisateurs d'interagir de manière sécurisée avec des documents et des dossiers, tout en garantissant que seuls les utilisateurs autorisés puissent effectuer certaines actions. De plus, il faut prendre en compte la complexité de la gestion des utilisateurs, des documents et des dossiers, tout en minimisant les erreurs et les risques d'opérations non autorisées. Comment développer une application Angular capable de relever ces défis tout en fournissant une expérience utilisateur fluide ?

3. Objectifs:

Les objectifs de ce projet incluent :

- L'authentification et Autorisation : Mettre en place un système d'authentification permettant aux utilisateurs de se connecter à l'application.
- La gestion des Utilisateurs : Permettre aux administrateurs de modifier (activer, désactiver) et de supprimer des utilisateurs.
- La gestion des Dossiers : Offrir la possibilité de créer des dossiers pour organiser les documents, permettant d'attribuer des documents à des dossiers spécifiques et de modifier les détails des dossiers, y compris leur contenu.
- La gestion des Documents par les utilisateurs : Permettre aux utilisateurs standard d'ajouter de nouveaux documents à un dossier spécifique avec leurs informations, de modifier les détails des documents existants, de supprimer des documents après confirmation et d'afficher la liste des documents.

-
- La gestion des Erreurs : Implémenter des messages d'erreur informatifs en cas d'opérations infructueuses. Gérer les exceptions et les erreurs côté client pour assurer une expérience utilisateur fluide.

4. Conclusion:

Ce chapitre a établi un cadre solide pour notre exploration en contextualisant le besoin d'une gestion de documents améliorée et en définissant clairement les objectifs du projet. En identifiant les défis inhérents à la gestion efficace des documents dans les organisations modernes, nous avons posé les bases pour développer une solution qui non seulement répond aux exigences fonctionnelles mais améliore également l'expérience utilisateur.

Chapitre 2: Modélisation

Dans ce chapitre, nous aborderons la phase essentielle de modélisation du projet en utilisant divers diagrammes, notamment les diagrammes de cas d'utilisation, qui mettent en lumière les interactions entre les acteurs et le système. De plus, nous utiliserons des diagrammes d'activité pour détailler les étapes séquentielles des processus clés.

UML, qui signifie "Unified Modeling Language" ou "langage de modélisation unifié", est un langage formel normalisé utilisé pour la modélisation orientée objet. L'une de ses caractéristiques clés est son indépendance totale vis-à-vis des langages de programmation, des domaines d'application et des processus, ce qui le rend universellement applicable.

UML joue un rôle essentiel en tant qu'outil de communication, permettant de représenter et de comprendre visuellement des solutions orientées objet. Sa notation graphique facilite l'expression visuelle des solutions, simplifiant ainsi la comparaison et l'évaluation des différentes approches. De plus, cette notation graphique contribue à réduire les ambiguïtés et les malentendus.

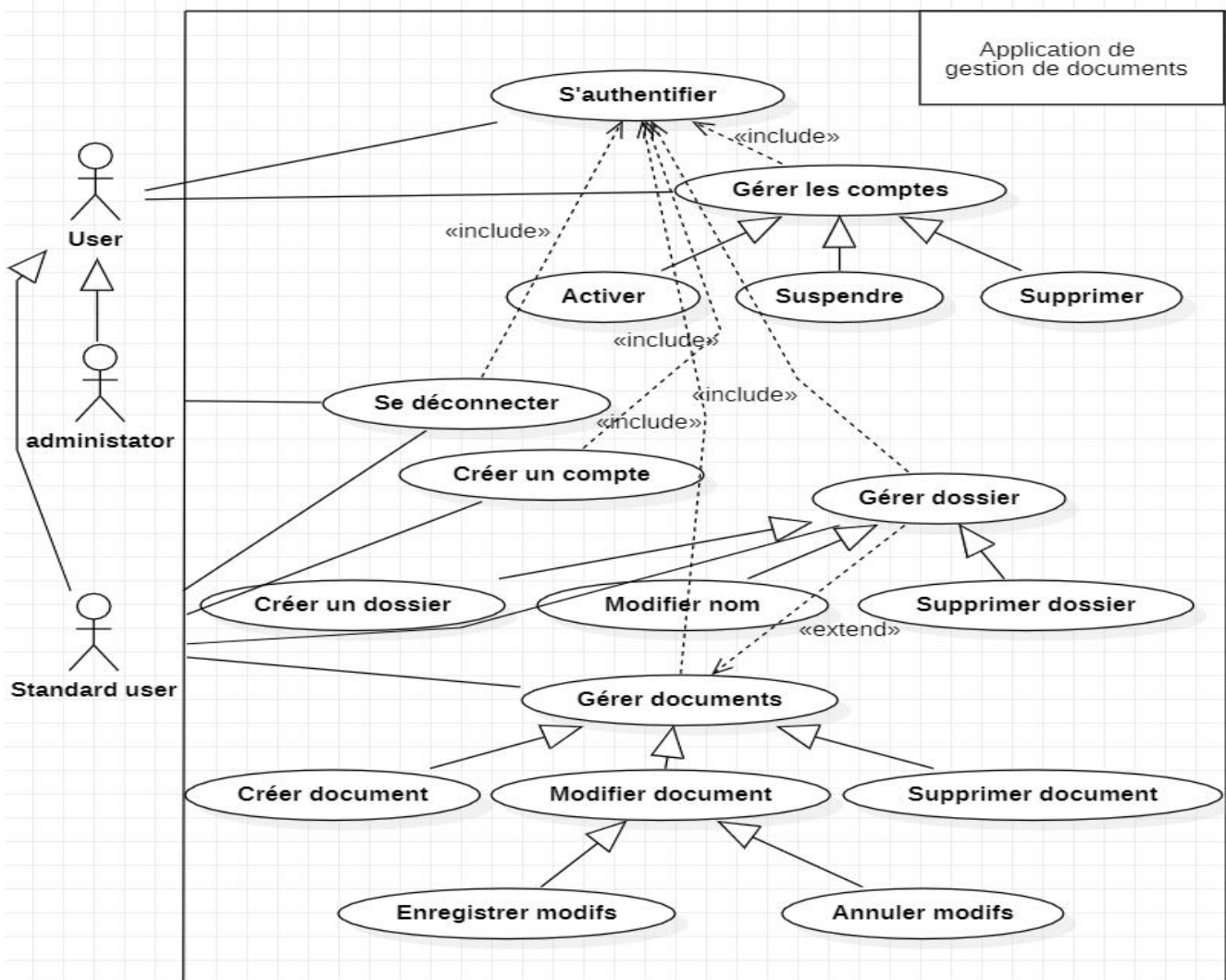
Dans le cadre de notre projet, nous avons opté pour deux types de diagrammes UML spécifiques : le diagramme de cas d'utilisation et le diagramme de séquence. Ces diagrammes sont des outils essentiels pour représenter et analyser les interactions entre les acteurs et le système, ainsi que la séquence des actions au sein du système, respectivement.

1. Diagramme de cas d'utilisation:

Le diagramme de cas d'utilisation est un outil de modélisation qui permet de représenter les interactions entre les acteurs, tels que les utilisateurs et les systèmes externes, ainsi que les différentes fonctionnalités offertes par un système.

Son objectif principal est de mettre en avant les objectifs fonctionnels du système, plutôt que de se concentrer sur les détails techniques de sa réalisation.

En d'autres termes, il sert à visualiser les scénarios d'utilisation et les relations entre les acteurs et les cas d'utilisation, afin de mieux



comprendre les besoins et les fonctionnalités essentielles du système.

Description des cas d'utilisation par utilisateur:

Administrateur :

- S'authentifier : L'administrateur doit se connecter à l'application.
- Gérer les comptes : L'administrateur peut activer, suspendre ou supprimer des comptes d'utilisateurs :

-Activer : Permettre à un compte utilisateur d'accéder au système.

-
- Suspendre : Temporairement désactiver l'accès d'un compte utilisateur.
 - Supprimer : Effacer complètement un compte utilisateur du système.
 - Se déconnecter : L'administrateur peut se déconnecter de l'application.

Utilisateur standard :

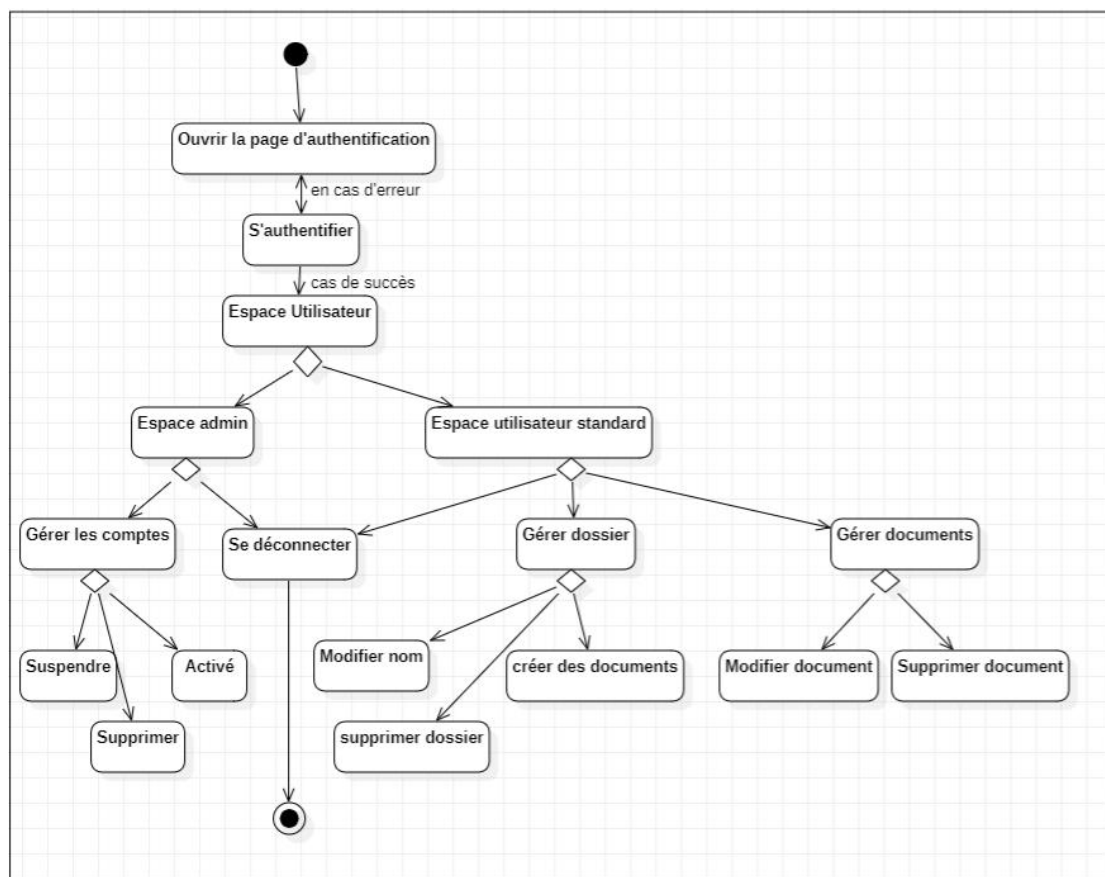
- S'authentifier : L'utilisateur standard doit se connecter à l'application.
- Créer un compte : L'utilisateur standard peut créer un nouveau compte utilisateur.
- Créer un dossier : L'utilisateur peut créer un nouveau dossier pour organiser les documents.
- Gérer dossier : L'utilisateur peut gérer les dossiers existants :
 - Modifier nom : Changer le nom d'un dossier existant.
 - Supprimer dossier : Effacer un dossier existant.
- Gérer documents : L'utilisateur peut effectuer des actions sur les documents.
 - Créer document : Créer un nouveau document dans un dossier.
 - Modifier document : Apporter des modifications à un document existant.
 - Supprimer document : Effacer un document existant.
 - Enregistrer modification: Sauvegarder les modifications apportées à un document.
 - Annuler modification: Annuler les dernières modifications non enregistrées sur un document.

- Se déconnecter : L'utilisateur standard peut se déconnecter de l'application.

2. Diagramme d'activités:

Le diagramme d'activités est un type de diagramme UML utilisé pour modéliser le flux de contrôle ou le flux d'activités au sein d'un système, d'un processus ou d'un cas d'utilisation.

Il représente graphiquement les étapes et les décisions d'un processus ou d'une activité, montrant comment les différents éléments interagissent entre eux.



2. Conclusion:

La modélisation, à l'aide de diagrammes UML, a joué un rôle crucial dans la clarification des exigences et des fonctionnalités de notre application. Cette approche nous a permis de visualiser les interactions entre les utilisateurs et le système, facilitant une compréhension commune des processus clés et des fonctionnalités. Les diagrammes de cas d'utilisation et d'activité ont été essentiels pour détailler le comportement attendu de l'application, assurant ainsi que le développement aligne étroitement avec les besoins des utilisateurs.

Chapitre 3 : Outils et Technologies :

La conception et le développement d'une application de gestion de documents nécessitent une palette d'outils et de technologies soigneusement sélectionnée. Dans ce chapitre, nous plongeons dans le cœur technique du projet, dévoilant les langages de programmation, les frameworks et les systèmes de bases de données qui donnent vie à notre application. Chaque choix technologique est stratégique, visant à optimiser la performance, la sécurité et l'expérience utilisateur.

1. Front-End :

Angular Framework :

- Angular CLI : Pour initialiser, développer et maintenir des applications Angular.
- TypeScript : Langage de programmation principal pour le développement Angular.
- Angular Components : Pour créer des composants modulaires pour différentes parties de l'application (par exemple, composants pour la gestion des utilisateurs, la gestion des documents, etc.).
- Angular Services : Pour gérer la logique métier et l'accès aux données.
- Angular Router : Pour la navigation et la gestion des routes dans l'application.
- Angular Forms : Pour gérer les formulaires et la validation des données utilisateur.



Authentification et Autorisation :



- Angular Guard : Pour la protection des routes et la gestion des autorisations basées sur les rôles.
- JWT (JSON Web Tokens) : Pour sécuriser les échanges entre le client et le serveur après l'authentification.

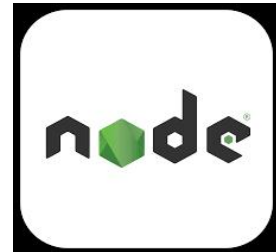
Interface Utilisateur (UI) :

- HTML5/CSS3 : Pour la structure et le style de l'application.

2. Back-End :

API REST :

- Node.js : un environnement d'exécution pour le JavaScript côté serveur, ce qui permet aux développeurs d'utiliser JavaScript pour écrire des scripts serveur.
- Express.js : Cadre pour créer des APIs web.



Base de Données :



- MongoDB : Système de gestion de base de données NoSQL pour stocker les informations des documents et des dossiers.
- Mongoose : Bibliothèque pour la modélisation des objets MongoDB pour Node.js.

Sécurité et Authentification:

- Authentification :

bcrypt ✧ bcrypt.js : Pour hasher et sécuriser les mots de passe.



✧ CORS (Cross-Origin Resource Sharing):

Un package middleware pour Express qui peut être utilisé pour activer CORS avec diverses options. CORS est un mécanisme qui permet à des ressources restreintes sur une page web d'être récupérées par un autre domaine extérieur au domaine qui a servi la première page web. C'est une pratique de sécurité essentielle pour empêcher les requêtes malveillantes de domaines non autorisés

- Gestion des Erreurs :

- ✧ Middleware Express custom : Pour gérer les erreurs côté serveur.
- ✧ Angular HttpClient interceptors : Pour intercepter et gérer les erreurs côté client.



3. Développement et Modélisation:



débogage.

WebStorm: Un IDE puissant spécialement conçu pour le développement JavaScript et Angular. Il offre des fonctionnalités avancées telles que l'analyse de code, le refactoring intelligent, une navigation rapide du projet et une riche interface de

StarUML : Utilisé pour la création de diagrammes UML nécessaires à la documentation de la conception architecturale de l'application. Il permet de modéliser les relations entre les composants de l'application, les structures de données et les interactions utilisateur.



4. Conclusion:

Ce chapitre a mis en lumière la sélection stratégique d'outils et technologies qui sous-tendent le développement de notre application. En choisissant Angular, Node.js, Express.js, et MongoDB, nous avons formé une pile technologique robuste qui favorise la performance, la sécurité, et une expérience utilisateur exceptionnelle.

Chapitre 4 : Réalisation

Dans ce chapitre, nous présentons la réalisation finale de notre application de gestion de documents.

À travers une série de captures d'écran, nous illustrerons les fonctionnalités principales et l'interface utilisateur, soulignant comment les objectifs définis ont été atteints. Nous discuterons également l'expérience utilisateur, en partageant des insights sur la manière dont l'application répond aux besoins des administrateurs et des utilisateurs standards.

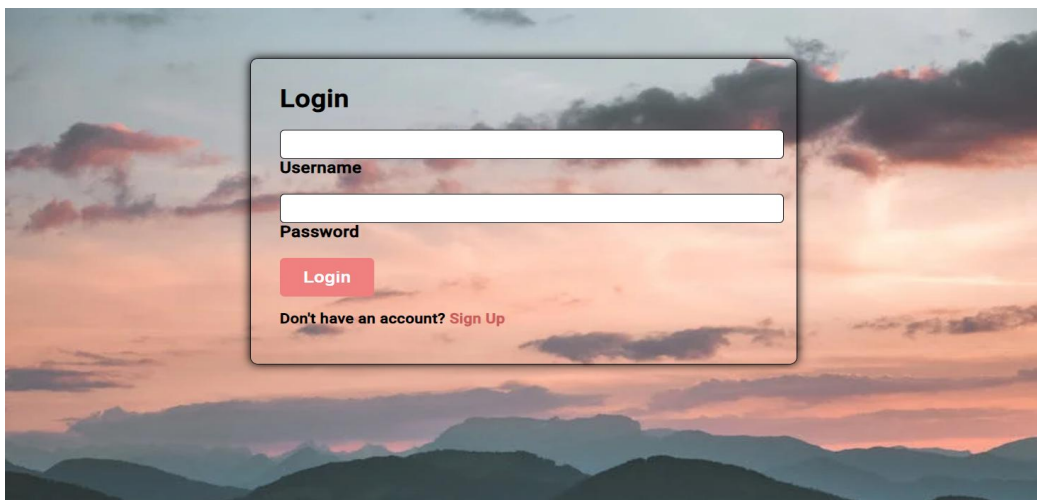
Ce chapitre vise à offrir une vue d'ensemble claire de l'application en action, démontrant son efficacité et son utilité dans le contexte de la gestion de documents.

1. Authentification:

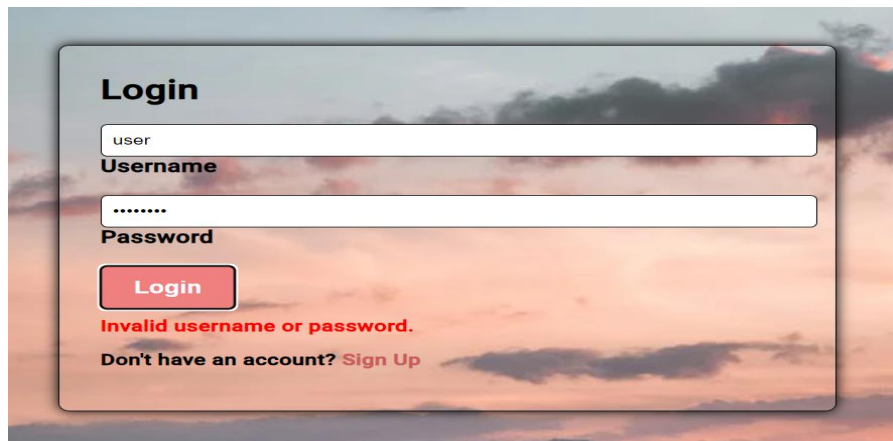
Lorsque notre application est lancée, la première étape requise est la page de connexion . Il est impératif pour les utilisateurs de se connecter afin de naviguer vers l'interface appropriée en fonction de leur rôle.

a) Connexion:

L'écran de connexion, tel qu'illustré par la capture d'écran, sollicite les identifiants de l'utilisateur, à savoir le nom d'utilisateur et le mot de passe.



Si les détails fournis ne correspondent pas à un compte existant ou si le mot de passe saisi est incorrect, un message d'erreur sera affiché pour informer l'utilisateur.



Login

user
Username

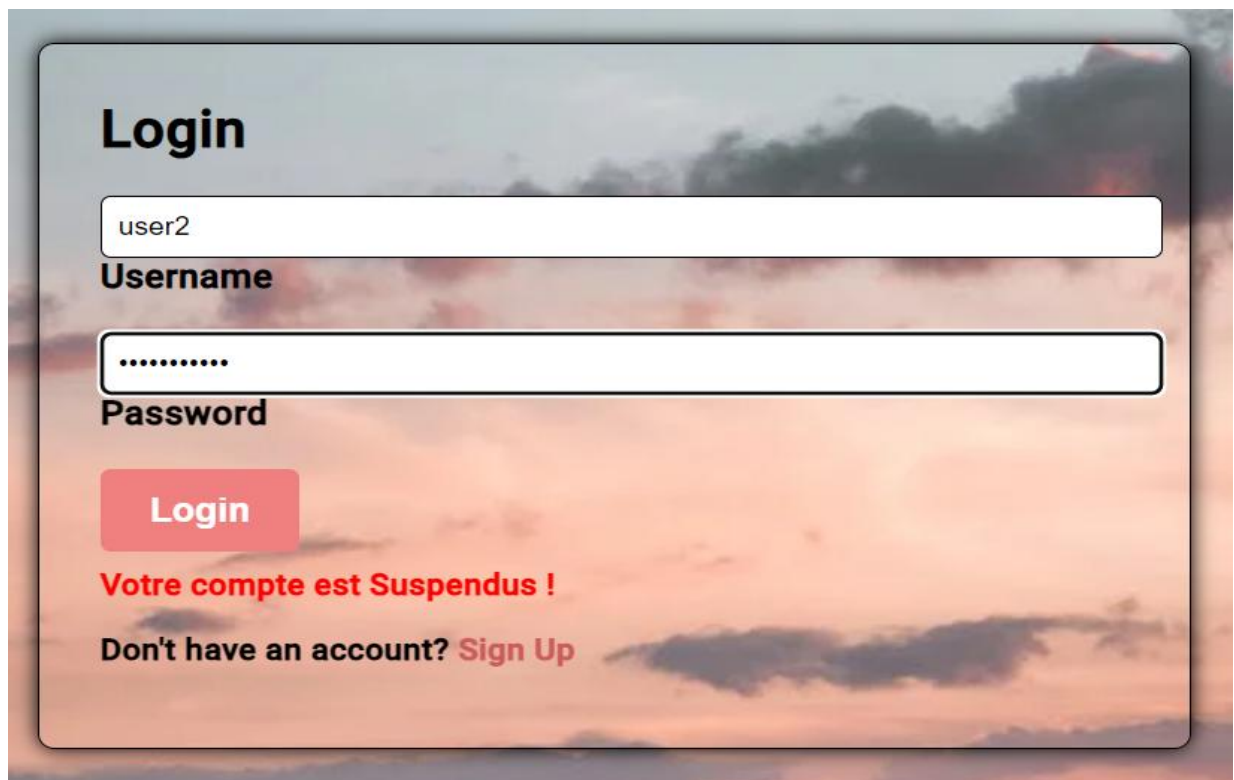
.....
Password

Login

Invalid username or password.

Don't have an account? [Sign Up](#)

Si l'utilisateur est suspendu , il ne peut pas accéder à la plateforme et un message d'erreur sera affiché :



Login

user2
Username

.....
Password

Login

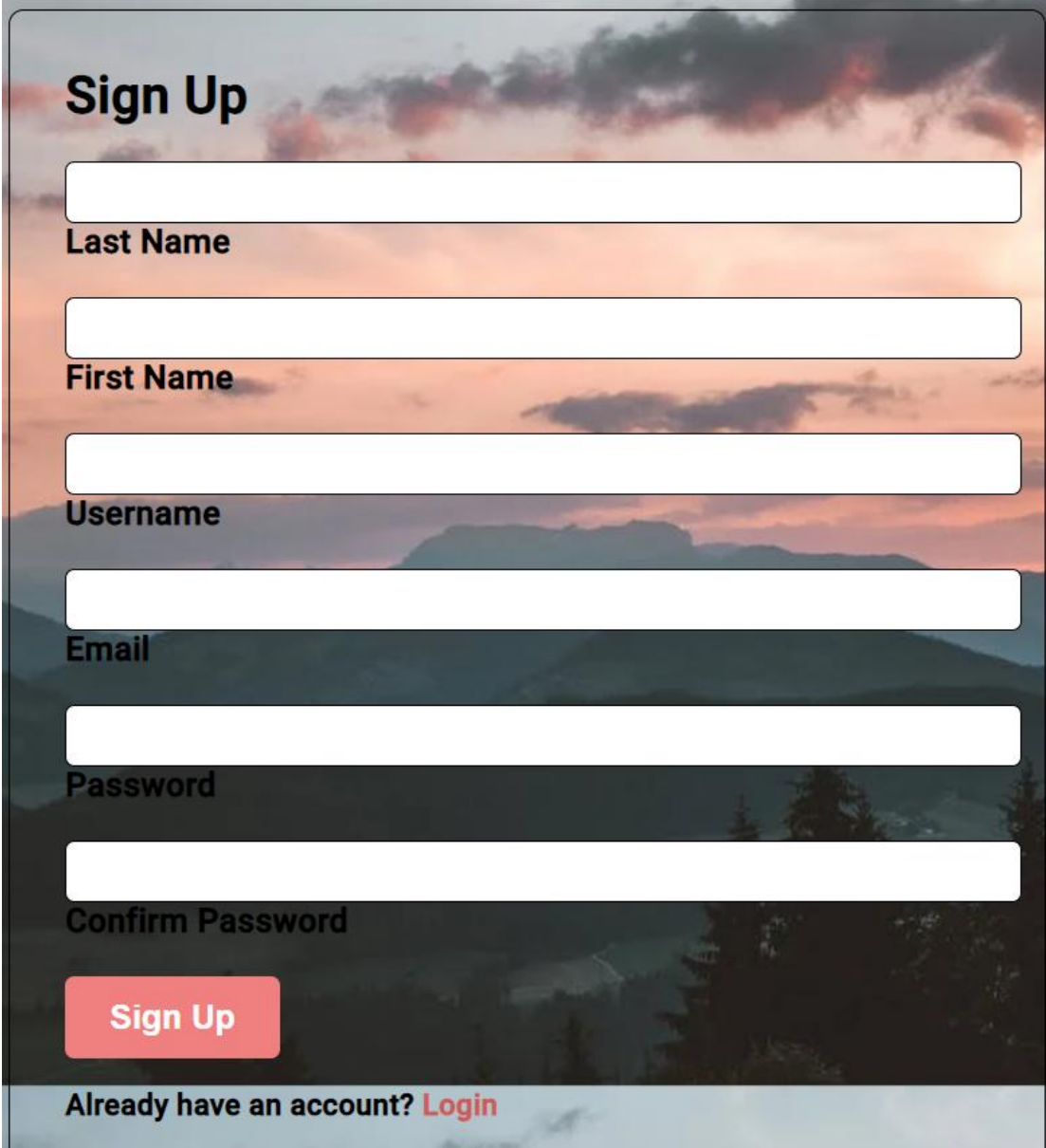
Votre compte est Suspendus !

Don't have an account? [Sign Up](#)

b)Inscription:

Si l'utilisateur ne possède pas de compte, il a la possibilité de s'en créer un en sélectionnant l'option "Sign Up". Le processus d'inscription requiert la saisie de son nom, prénom, nom d'utilisateur, une adresse e-mail valide et la création d'un mot de passe conforme aux critères de sécurité, incluant lettres, chiffres,

caractères spéciaux et une longueur minimale de huit caractères. Une seconde validation du mot de passe est demandée pour confirmer son exactitude.



Sign Up

Last Name

First Name

Username

Email

Password

Confirm Password

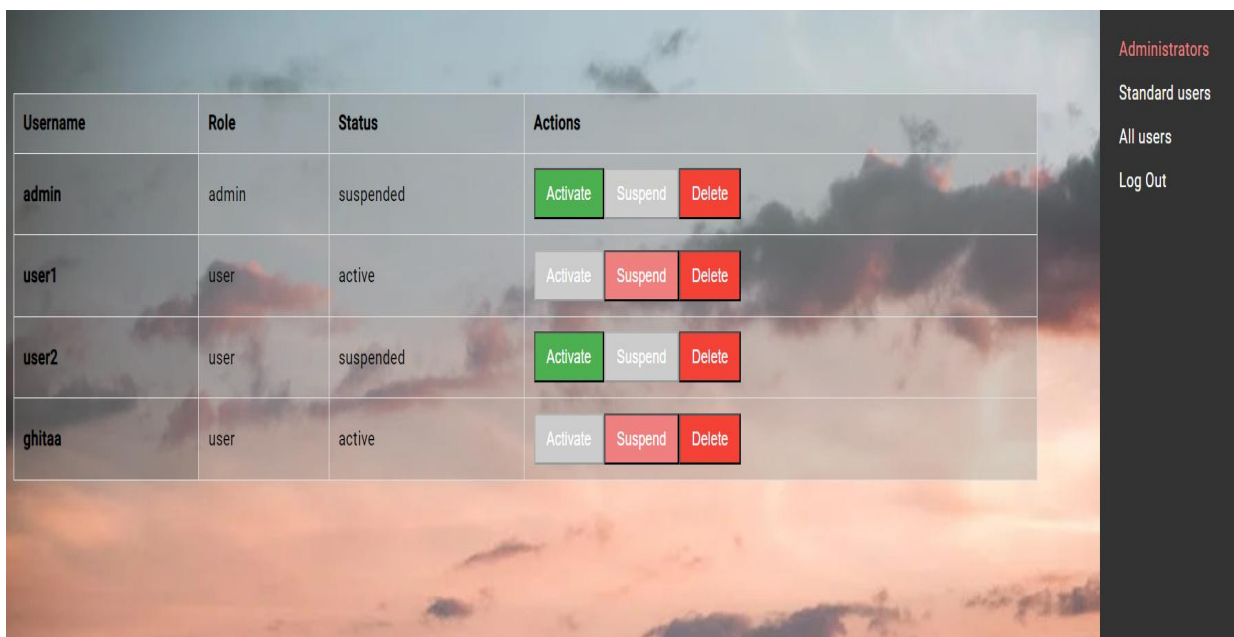
Already have an account? [Login](#)

2. Interface Administrateur :

L'interface administrateur présente un tableau clair et structuré, dédié à la supervision des utilisateurs. chaque ligne représente un utilisateur distinct. Les colonnes affichent le nom d'utilisateur, le rôle attribué (administrateur ou utilisateur standard), et l'état actuel de l'utilisateur, indiquant s'ils sont actifs ou suspendus.

L'administrateur a la capacité de modifier l'état de chaque compte avec des options pour activer, suspendre, ou supprimer des utilisateurs, facilitant ainsi la gestion directe des accès et des permissions.

Le menu de navigation permet à l'administrateur de filtrer les utilisateurs : les utilisateurs standards ou tous les utilisateurs, et de se déconnecter pour sortir en toute sécurité de la session d'administration.



The screenshot displays the Admin Interface. It features a table with columns for Username, Role, Status, and Actions. The table lists four users: 'admin' (suspended), 'user1' (active), 'user2' (suspended), and 'ghitaa' (active). Each user has three action buttons: 'Activate' (green), 'Suspend' (grey), and 'Delete' (red). To the right of the table is a dark sidebar with a navigation menu containing 'Administrators' (highlighted in red), 'Standard users', 'All users', and 'Log Out'.

Username	Role	Status	Actions
admin	admin	suspended	<button>Activate</button> <button>Suspend</button> <button>Delete</button>
user1	user	active	<button>Activate</button> <button>Suspend</button> <button>Delete</button>
user2	user	suspended	<button>Activate</button> <button>Suspend</button> <button>Delete</button>
ghitaa	user	active	<button>Activate</button> <button>Suspend</button> <button>Delete</button>

Administrators

Standard users

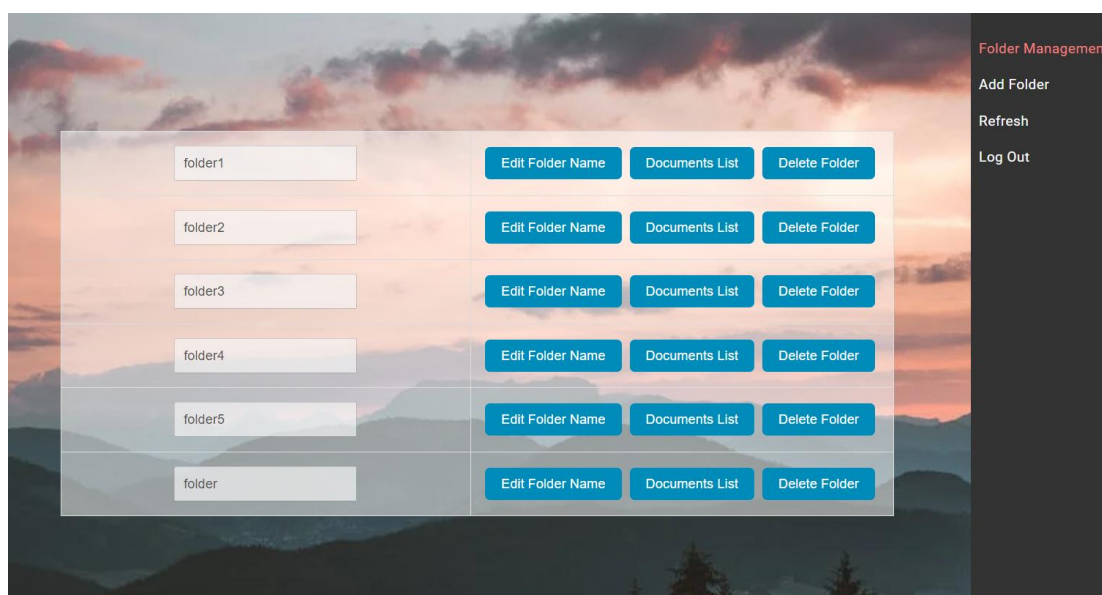
All users

Log Out

3. Interface utilisateur standard :

a) Liste des dossiers :

Après s'être connecté, l'utilisateur est redirigé vers une page qui affiche la liste des dossiers déjà créés. Cette page est équipée de boutons permettant de renommer ou de supprimer chaque dossier, ainsi que d'une option pour examiner les documents qu'il contient. De plus, l'interface propose un bouton pour ajouter de nouveaux dossiers, rafraîchir la vue actuelle ou se déconnecter de l'application.



b) Ajouter nouveau dossier :

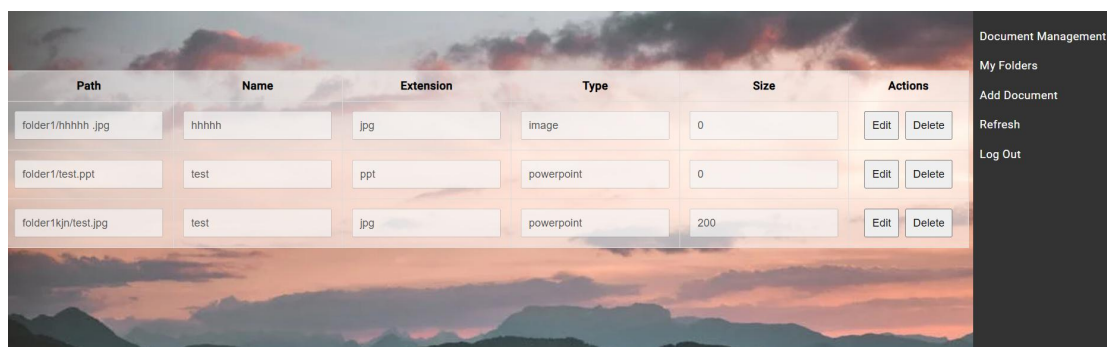
Lors de la sélection de l'option "Add Folder", l'utilisateur est dirigé vers une page où il peut initier l'ajout d'un nouveau dossier. Il lui suffit d'entrer le nom souhaité pour le dossier dans le champ prévu à cet effet et de valider en cliquant sur le bouton de confirmation pour compléter l'ajout du dossier.

The screenshot shows a modal window titled 'Add Folder' with a close button (X) in the top right corner. Inside the modal, there is a label 'Folder Name' followed by a text input field. Below the input field is a large red button with the text 'Add folder' in white.

c) Liste des documents :

Lorsque l'utilisateur choisit l'option "Documents List", une page s'ouvre pour afficher une liste détaillée de tous les documents disponibles. Chaque document est accompagné de ses informations essentielles, telles que le chemin, le nom, l'extension, le type et la taille. En plus de ces détails, des boutons sont mis à disposition pour permettre la modification ou la suppression d'un document spécifique.

Dans la barre latérale, l'utilisateur trouve un ensemble d'options pratiques. Il peut ajouter un nouveau document au dossier en cours, actualiser la page pour obtenir les dernières informations, se déconnecter de son compte ou retourner à la liste des dossiers pour une navigation fluide au sein de l'application.



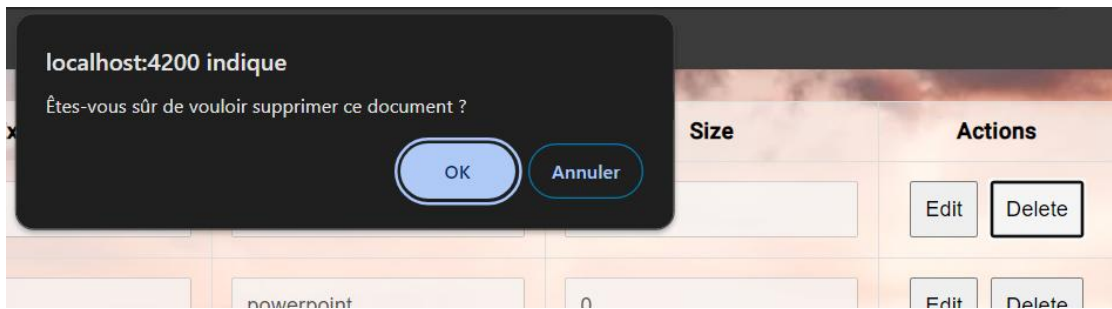
Path	Name	Extension	Type	Size	Actions
folder1/hhhhh.jpg	hhhhh	jpg	image	0	Edit Delete
folder1/test.ppt	test	ppt	powerpoint	0	Edit Delete
folder1/kjn/test.jpg	test	jpg	powerpoint	200	Edit Delete

Lors de la procédure de modification, l'utilisateur est présenté avec une option pour confirmer s'il souhaite réellement effectuer des modifications ou revenir aux données précédentes.



folder1/hhhhh.jpg	hhhhh	jpg	image	0	Save	Cancel
-------------------	-------	-----	-------	---	------	--------

En ce qui concerne la suppression d'un élément, il est nécessaire de confirmer l'intention de suppression avant que celle-ci ne soit exécutée.



d) Ajouter un nouveau document :

Lorsque l'utilisateur sélectionne l'option "Add Document", une fenêtre s'ouvre, lui permettant de spécifier les détails essentiels du document qu'il souhaite ajouter. Il doit fournir le nom du document, sa taille, son extension et son type, puis confirmer l'ajout en appuyant sur le bouton de validation.

4. Déconnexion :

L'option de déconnexion est accessible à tous les utilisateurs de l'application. Une fois déconnecté, l'utilisateur est automatiquement redirigé vers la page de connexion pour une nouvelle session.

5. Conclusion :

En résumé, ce chapitre a offert une vue d'ensemble claire de l'application en action, démontrant son efficacité, sa sécurité et son utilité dans le contexte de la gestion de documents. Il s'agit d'une étape clé de notre rapport, mettant en avant le résultat concret de notre travail acharné pour répondre aux besoins des utilisateurs et assurer une gestion de documents fluide et sécurisée.

Chapitre : Conclusion :

Ce rapport a exploré en détail le projet de développement d'une application de gestion de documents utilisateur construite en utilisant le framework Angular, Node.js, Express.js et MongoDB. Nous avons plongé dans les différents aspects du projet, de sa conception à sa réalisation, en mettant en lumière les choix architecturaux, les fonctionnalités clés et les considérations de sécurité.

L'introduction générale a exposé le contexte de la gestion de documents dans un monde de plus en plus numérique, soulignant l'importance de stocker et d'organiser efficacement l'information. Le rapport a ensuite décrit les objectifs principaux de l'application, à savoir répondre aux besoins des administrateurs et des utilisateurs standards en offrant une gestion fluide des documents et des dossiers.

Le chapitre de modélisation a utilisé des diagrammes UML, notamment les diagrammes de cas d'utilisation et d'activité, pour visualiser les interactions entre les acteurs et le système, ainsi que les processus clés de l'application. Cela a permis de clarifier les exigences et de guider le développement.

Dans le chapitre sur les outils et les technologies, nous avons détaillé les choix technologiques, des langages de programmation aux bases de données, qui ont contribué à la création d'une application performante et sécurisée. Les outils tels que Angular, Node.js et MongoDB ont été sélectionnés pour leur adaptabilité et leur efficacité.

Le chapitre de réalisation a mis en avant les fonctionnalités concrètes de l'application, de l'authentification à la gestion des utilisateurs, des dossiers et des documents. L'interface utilisateur a été présentée à travers des captures d'écran, illustrant comment les objectifs ont été atteints pour une expérience utilisateur fluide et sécurisée.

Livable :

lien de la video :

<https://www.youtube.com/watch?v=Ak0Ic8kHmKw&feature=youtu.be>