PROJET ANNUEL

BERDRIN Tristan

DELFINO Alexandre

KINSIONA MFINDA Clément

Une image contenant texte, Police, Graphique, logo

Description générée automatiquement

**Résumé Exécutif**

Le projet vise à développer une application intégrant une API et divers clients, sur le thème de la gestion d'association. L'objectif principal du projet est de concevoir une API en NodeJS utilisant un système de gestion de base de données.  
Le projet inclut aussi la conception d’un client web React destiné aux utilisateurs finaux, un client web pour l'administration back office, ainsi qu'un client Java graphique pour la planification des activités des membres salariés de l'association.

Le projet doit inclure la gestion des assemblées générales, un mécanisme de vote en ligne, un mécanisme de gestion des dons, des cotisations et des rappels. Une gestion électronique des documents doit être mise en place, permettant de traiter, exploiter et archiver les documents de manière électronique.   
  
  
AJOUTER JAVA ICI

Les résultats attendus incluent une application entièrement conteneurisée et versionnée sur GitHub, avec l'utilisation de Trello.

**Description du Projet**

Le projet consiste à développer un site web pour la gestion d'une association. L'objectif principal est de créer une plateforme en ligne qui facilite la gestion des membres, des cotisations, des actions et des activités de l'association, tout en offrant des fonctionnalités avancées pour améliorer l'efficacité de l'organisation.  
  
L’application web est destinée à toute association qui nécessite une solution intégrée pour gérer ses opérations quotidiennes. Le public cible inclut les administrateurs de l'association, les membres, ainsi que les bénéficiaires et partenaires. L'objectif est de fournir un outil pour la gestion des différentes facettes de l'association, de la gestion des membres aux assemblées générales, en passant par la collecte de dons et la gestion documentaire.

**Architecture Technique**

Les interfaces utilisateur comprennent un client web développé en React avec TypeScript.

La base de données repose sur MySQL. Elle est utilisée pour stocker les données de l'association, telles que les informations des membres, les cotisations, les actions et les activités. Pour la gestion de la base de données via une interface web, nous employons phpMyAdmin.

Les services backend sont développés avec Node.js et Express.js, un framework pour Node.js, est utilisé pour gérer les requêtes HTTP et les routes de l'application. Enfin, nous utilisons Joi qui est une bibliothèque de validation de requêtes qui permet de valider les données entrantes sur le serveur backend, assurant que les requêtes respectent les schémas définis.

**Technologies Utilisées**

La partie web a été faite à l’aide de React TS, HTML,CSS.  
La base de données a été faites en SQL et peut être gérée depuis PhpMyAdmin.  
L’API et la gestion de la base de données ont été faites à l’aide de Node JS et Express JS.

**Conception de la Base de Données**

La conception de la base de données pour le projet est structurée pour répondre aux besoins spécifiques de gestion des assemblées générales, des documents, des rôles des utilisateurs, des sondages, des votes et des réponses aux votes.   
La table ag est utilisée pour enregistrer les détails des assemblées générales, y compris le titre, la date, l'heure et l'ordre du jour, ainsi que le créateur de l'assemblée. La table ag\_email associe les emails des participants aux assemblées générales spécifiques.

La table document sert à gérer les documents de l'association, stockant des informations telles que la description, l'URL du fichier, le titre et la catégorie du document. Les modifications apportées aux documents sont suivies dans la table document\_log, qui enregistre les descriptions des changements et les horodatages.

La table role définit les rôles des utilisateurs, permettant de gérer les permissions et les accès au sein de l'association. La table user enregistre les informations des utilisateurs, y compris leurs identifiants, emails, mots de passe, et leur rôle associé.

Pour les sondages, la table survey enregistre les détails des sondages, y compris le titre, la date de création, le créateur et la date limite. Les questions des sondages sont stockées dans la table survey\_question, tandis que les réponses sont enregistrées dans la table survey\_response.

Les tokens de sécurité pour les sessions des utilisateurs sont gérés dans la table token. Les votes, y compris les détails des titres, modes de vote, et majorités requises, sont enregistrés dans la table vote. Les options de vote sont stockées dans la table vote\_option, et les questions associées aux votes dans la table vote\_question. Enfin, les réponses aux votes sont enregistrées dans la table vote\_response.

**Plan de Sécurité**

Pour assurer la sécurité de notre site web et protéger contre diverses menaces, plusieurs mesures de sécurité ont été mises en place. Ces mesures couvrent l'authentification, l'autorisation, ainsi que la protection contre les attaques courantes.  
Pour l'authentification, nous utilisons JSON Web Tokens pour gérer l'authentification basée sur des tokens. Les utilisateurs reçoivent un JWT après une connexion réussie, qui est ensuite utilisé pour authentifier les requêtes suivantes.   
Concernant l'autorisation, nous implémentons un système de rôles et permissions pour restreindre l'accès à certaines parties de l'application.

**Plan de Déploiement**

**Gestion des Performances**

**Tests et Validation**

**Documentation**

**Plan de Maintenance**

**Budget et Ressources**