

PROJET PROFESSIONNEL L 2023



**Share
Event**

TABLE DES MATIÈRES

INTRODUCTION

- 1.1 CONTEXTE ET MOTIVATION**
- 1.2 OBJECTIFS DE L'APPLICATION SHAREEVENT**
- 1.3 STRUCTURE DU DOSSIER**

ÉTAT DE L'ART

- 2.1 APPLICATIONS SIMILAIRES EXISTANTES**
- 2.2 TECHNOLOGIES UTILISÉES**
- 2.3 AVANTAGES ET INCONVÉNIENTS DES TECHNOLOGIES CHOISIES**

CONCEPTION DE L'APPLICATION

- 3.1 SPÉCIFICATIONS FONCTIONNELLES ET NON-FONCTIONNELLES**
- 3.2 ARCHITECTURE DE L'APPLICATION**
- 3.3 CONCEPTION DE L'INTERFACE UTILISATEUR**
- 3.4 CONCEPTION BASE DE DONNÉE METHODE MERISE**

DÉVELOPPEMENT DE LA PARTIE FRONT-END

- 4.1 PRÉSENTATION DE REACT NATIVE**
- 4.2 IMPLÉMENTATION DES FONCTIONNALITÉS DE L'APPLICATION**
- 4.3 GESTION DE L'AUTHENTIFICATION ET DES AUTORISATIONS**

DÉVELOPPEMENT DE LA PARTIE BACK-END

- 5.1 INTRODUCTION À NESTJS**
- 5.2 MODÉLISATION DE LA BASE DE DONNÉES AVEC TYPEORM**
- 5.3 IMPLÉMENTATION DES FONCTIONNALITÉS DU SERVEUR**

TESTS ET VALIDATION

- 6.1 STRATÉGIE DE TESTS**
- 6.2 TESTS UNITAIRES ET TESTS D'INTÉGRATION**
- 6.3 VALIDATION DE L'APPLICATION AVEC DES UTILISATEURS RÉELS**

DISCUSSION ET AMÉLIORATIONS FUTURES

- 7.1 ÉVALUATION DES RÉSULTATS**
- 7.2 LIMITATIONS ET DÉFIS RENCONTRÉS**
- 7.3 PERSPECTIVES D'AMÉLIORATION**

CONCLUSION

- 8.1 RÉSUMÉ DES CONTRIBUTIONS**
- 8.2 RETOUR SUR L'EXPÉRIENCE DE DÉVELOPPEMENT**
- 8.3 CONCLUSION GÉNÉRALE**

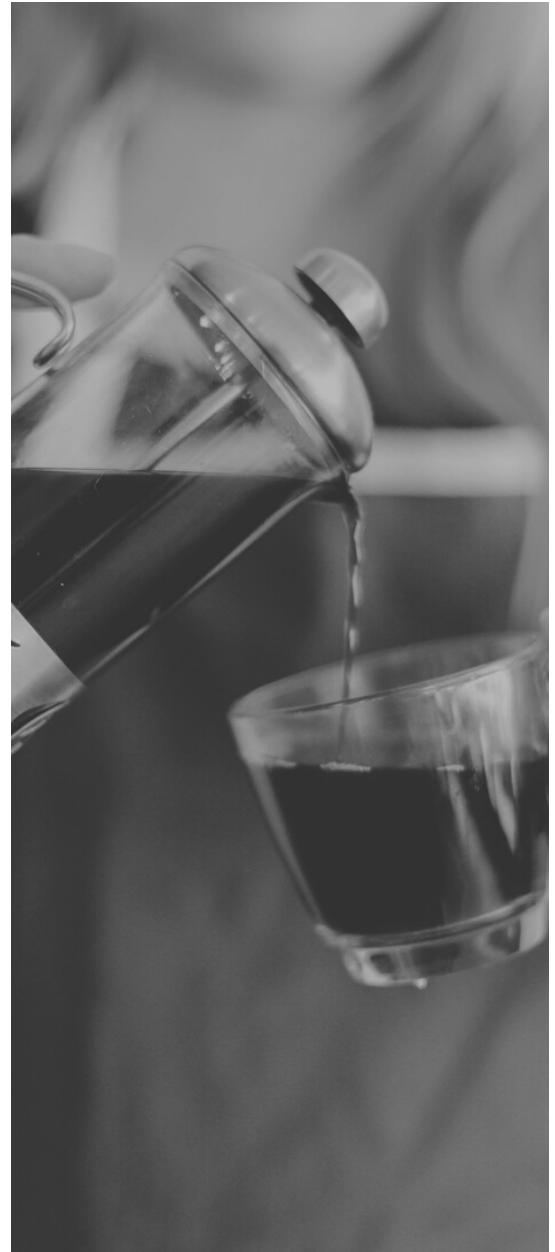
INTRODUCTION

Contexte et motivation

DANS UN MONDE DE PLUS EN PLUS CONNECTÉ ET OÙ LES ÉVÉNEMENTS SOCIAUX JOUENT UN RÔLE CENTRAL DANS NOS VIES, IL EST DEVENU ESSENTIEL DE DISPOSER D'UNE PLATEFORME PRATIQUE POUR CRÉER ET PARTAGER DES ÉVÉNEMENTS.

DE PLUS, LORS DE CES ÉVÉNEMENTS, LES PARTICIPANTS PRENNENT SOUVENT DE NOMBREUSES PHOTOS QU'ILS SOUHAITENT PARTAGER AVEC LES AUTRES PARTICIPANTS. CEPENDANT, IL PEUT ÊTRE DIFFICILE DE CENTRALISER TOUTES CES PHOTOS EN UN SEUL ENDROIT.

C'EST DANS CE CONTEXTE QUE L'APPLICATION SHAREEVENT A ÉTÉ DÉVELOPPÉE.



OBJECTIFS DE L'APPLICATION SHAREEVENT :

L'OBJECTIF PRINCIPAL DE SHAREEVENT EST DE SIMPLIFIER LA CRÉATION ET LE PARTAGE D'ÉVÉNEMENTS EN FOURNISSANT UNE PLATEFORME CONVIVIALE ET INTUITIVE. EN UTILISANT SHAREEVENT, LES UTILISATEURS PEUVENT CRÉER DES ÉVÉNEMENTS, INVITER D'AUTRES PARTICIPANTS ET PARTAGER FACILEMENT LES PHOTOS PRISES LORS DE CES ÉVÉNEMENTS. L'APPLICATION VISE À OFFRIR UNE EXPÉRIENCE UTILISATEUR FLUIDE ET À RÉSOUDRE LE PROBLÈME DE DISPERSION DES PHOTOS SUR DIFFÉRENTES PLATEFORMES DE MÉDIAS SOCIAUX.

STRUCTURE DU DOSSIER

CE DOSSIER EST STRUCTURÉ DE MANIÈRE À FOURNIR UNE COMPRÉHENSION COMPLÈTE DU DÉVELOPPEMENT DE L'APPLICATION SHAREEVENT. IL SE DIVISE EN PLUSIEURS SECTIONS CLÉS :

L'ÉTAT DU MARCHÉ EXAMINE LES APPLICATIONS SIMILAIRES EXISTANTES ET LES TECHNOLOGIES UTILISÉES DANS LE DÉVELOPPEMENT DE SHAREEVENT.

LA CONCEPTION DE L'APPLICATION PRÉSENTE LES SPÉCIFICATIONS FONCTIONNELLES ET NON-FONCTIONNELLES, L'ARCHITECTURE DE L'APPLICATION ET LA CONCEPTION DE L'INTERFACE UTILISATEUR.

LE DÉVELOPPEMENT DE LA PARTIE FRONT-END DÉCRIT L'UTILISATION DE REACT NATIVE POUR DÉVELOPPER L'INTERFACE MOBILE DE SHAREEVENT, Y COMPRIS L'IMPLÉMENTATION DES FONCTIONNALITÉS ET LA GESTION DE L'AUTHENTIFICATION ET DES AUTORISATIONS.

LE DÉVELOPPEMENT DE LA PARTIE BACK-END EXPLORE L'UTILISATION DE NESTJS AVEC TYPEORM POUR METTRE EN PLACE LE SERVEUR DE SHAREEVENT ET IMPLÉMENTER LES FONCTIONNALITÉS DU BACK-END.

LES TESTS ET LA VALIDATION DÉCRIVENT LA STRATÉGIE DE TESTS UTILISÉE, Y COMPRIS LES TESTS UNITAIRES, LES TESTS D'INTÉGRATION ET LA VALIDATION DE L'APPLICATION AVEC DES UTILISATEURS RÉELS.

LA DISCUSSION ET LES AMÉLIORATIONS FUTURES ÉVALUENT LES RÉSULTATS OBTENUS, IDENTIFIENT LES LIMITATIONS ET LES DÉFIS RENCONTRÉS, ET PROPOSENT DES PERSPECTIVES D'AMÉLIORATION POUR SHAREEVENT.

ENFIN, LA CONCLUSION RÉSUME LES CONTRIBUTIONS DU MÉMOIRE, PARTAGE LES RETOURS SUR L'EXPÉRIENCE DE DÉVELOPPEMENT ET PROPOSE UNE CONCLUSION GÉNÉRALE.

CE MÉMOIRE VISE À FOURNIR UN APERÇU COMPLET DU DÉVELOPPEMENT DE L'APPLICATION SHAREEVENT, EN METTANT L'ACCENT SUR LES ASPECTS TECHNIQUES, LES CHOIX DE CONCEPTION ET LES PERFORMANCES DE L'APPLICATION.

IL CONSTITUE UNE BASE SOLIDE POUR COMPRENDRE LE PROCESSUS DE DÉVELOPPEMENT D'UNE APPLICATION MOBILE UTILISANT REACT NATIVE ET NESTJS, AINSI QUE LES DÉFIS ET LES OPPORTUNITÉS LIÉS À LA CRÉATION D'UNE APPLICATION DE PARTAGE D'ÉVÉNEMENTS CENTRÉE SUR LES PHOTOS.

ETUDE DU MARCHÉ

Applications similaires existantes

AVANT DE DÉVELOPPER L'APPLICATION SHAREEVENT, IL EST IMPORTANT DE COMPRENDRE L'ÉTAT ACTUEL DU MARCHÉ DES APPLICATIONS DE PARTAGE D'ÉVÉNEMENTS ET DE CENTRALISATION DE PHOTOS. PLUSIEURS APPLICATIONS POPULAIRES OFFRENT DES FONCTIONNALITÉS SIMILAIRES, TELLES QUE "EVENTBRITE", "MEETUP" ET "FACEBOOK EVENTS".

CES APPLICATIONS PERMETTENT AUX UTILISATEURS DE CRÉER, DE GÉRER ET DE PARTAGER DES ÉVÉNEMENTS AVEC D'AUTRES PARTICIPANTS.

CEPENDANT, CES APPLICATIONS SE CONCENTRENT PRINCIPALEMENT SUR LA GESTION GÉNÉRALE DES ÉVÉNEMENTS ET NE PROPOSENT PAS UNE FONCTIONNALITÉ SPÉCIFIQUE POUR CENTRALISER LES PHOTOS PRISES LORS DE CES ÉVÉNEMENTS.

LES UTILISATEURS SONT SOUVENT CONTRAINTS DE PARTAGER LEURS PHOTOS SUR DIFFÉRENTES PLATEFORMES DE MÉDIAS SOCIAUX TELLES QUE FACEBOOK, INSTAGRAM OU WHATSAPP, CE QUI ENTRAÎNE UNE DISPERSION DES PHOTOS ET UNE DIFFICULTÉ POUR LES RETROUVER ULTÉRIEUREMENT.

2.2 TECHNOLOGIES UTILISÉES :

DANS LE CADRE DU DÉVELOPPEMENT DE SHAREEVENT, DEUX TECHNOLOGIES CLÉS ONT ÉTÉ UTILISÉES : REACT NATIVE POUR LA PARTIE FRONT-END MOBILE ET NESTJS AVEC TYPEORM POUR LA PARTIE BACK-END.

REACT NATIVE EST UN FRAMEWORK POPULAIRE QUI PERMET DE DÉVELOPPER DES APPLICATIONS MOBILES MULTIPLATEFORMES EN UTILISANT JAVASCRIPT. IL OFFRE UNE APPROCHE DE DÉVELOPPEMENT RAPIDE ET EFFICACE, EN PERMETTANT AUX DÉVELOPPEURS DE RÉUTILISER UNE GRANDE PARTIE DU CODE ENTRE LES PLATEFORMES IOS ET ANDROID. DE PLUS, REACT NATIVE FOURNIT UNE LARGE GAMME DE COMPOSANTS PRÉCONSTRUITS ET DE BIBLIOTHÈQUES TIERCES, CE QUI FACILITE LE DÉVELOPPEMENT D'INTERFACES UTILISATEUR ATTRAYANTES ET RÉACTIVES.

NESTJS EST UN FRAMEWORK BACK-END BASÉ SUR NODE.JS QUI UTILISE LES PRINCIPES DE L'ARCHITECTURE ORIENTÉE VERS LES SERVICES (SOA) ET DE LA PROGRAMMATION ORIENTÉE OBJET (POO). IL FACILITE LA CRÉATION D'APPLICATIONS SERVEUR ROBUSTES ET ÉVOLUTIVES EN FOURNISSANT UNE STRUCTURE BIEN DÉFINIE ET DES FONCTIONNALITÉS TELLES QUE LA GESTION DES ROUTES, L'INJECTION DE DÉPENDANCES ET LA GESTION DES REQUÊTES HTTP. TYPEORM, QUANT À LUI, EST UN ORM (OBJECT-RELATIONAL MAPPING) COMPATIBLE AVEC NESTJS, QUI PERMET DE MODÉLISER ET D'INTERAGIR AVEC LA BASE DE DONNÉES DE MANIÈRE INTUITIVE EN UTILISANT DES OBJETS ET DES REQUÊTES EN LANGAGE TYPESCRIPT.

EN CE QUI CONCERNE LE CHOIX DE NESTJS AVEC TYPEORM POUR LA PARTIE BACK-END, CES TECHNOLOGIES OFFRENT UNE STRUCTURE ET DES FONCTIONNALITÉS PUISSANTES POUR DÉVELOPPER UNE APPLICATION SERVEUR ROBUSTE. NESTJS FACILITE LA GESTION DES ROUTES, LA CRÉATION DE SERVICES MODULAIRES ET LA GESTION DES REQUÊTES HTTP. TYPEORM, EN TANT QU'ORM, SIMPLIFIE L'INTERACTION AVEC LA BASE DE DONNÉES EN UTILISANT DES OBJETS ET DES REQUÊTES EN LANGAGE TYPESCRIPT. CEPENDANT, LA COURBE D'APPRENTISSAGE INITIALE PEUT ÊTRE PLUS É

2.3 AVANTAGES ET INCONVÉNIENTS DES TECHNOLOGIES CHOISIES :

LE CHOIX DE REACT NATIVE POUR LA PARTIE FRONT-END MOBILE DE SHAREEVENT PRÉSENTE PLUSIEURS AVANTAGES SIGNIFICATIFS. TOUT D'ABORD, EN UTILISANT REACT NATIVE, IL EST POSSIBLE DE DÉVELOPPER UNE APPLICATION MOBILE MULTIPLATEFORME, CE QUI PERMET DE RÉDUIRE CONSIDÉRABLEMENT LES EFFORTS DE DÉVELOPPEMENT ET DE MAINTENANCE. LA RÉUTILISATION D'UNE GRANDE PARTIE DU CODE ENTRE LES PLATEFORMES IOS ET ANDROID ACCÉLÈRE LE PROCESSUS DE DÉVELOPPEMENT ET ASSURE UNE COHÉRENCE DANS L'EXPÉRIENCE UTILISATEUR. DE PLUS, LA VASTE BIBLIOTHÈQUE DE COMPOSANTS PRÉCONSTRUITS ET LES NOMBREUSES BIBLIOTHÈQUES TIERCES DISPONIBLES FACILITENT LA CRÉATION D'INTERFACES UTILISATEUR ATTRAYANTES ET RÉACTIVES. CELA PERMET AUX DÉVELOPPEURS DE BÉNÉFICIER D'UNE PRODUCTIVITÉ ACCRUE ET DE RÉDUIRE LES DÉLAIS DE DÉVELOPPEMENT.

CEPENDANT, L'UTILISATION DE REACT NATIVE PEUT PRÉSENTER QUELQUES INCONVÉNIENTS POTENTIELS. CERTAINES FONCTIONNALITÉS AVANCÉES SPÉCIFIQUES À CHAQUE PLATEFORME PEUVENT ÊTRE PLUS DIFFICILES À IMPLÉMENTER AVEC REACT NATIVE, NÉCESSITANT PARFOIS DES MODULES NATIFS PERSONNALISÉS. DE PLUS, LES PERFORMANCES DE L'APPLICATION PEUVENT ÊTRE LÉGÈREMENT INFÉRIEURES À CELLES D'UNE APPLICATION ENTIÈREMENT NATIVE, BIEN QUE LES AMÉLIORATIONS CONSTANTES DE REACT NATIVE CONTRIBUENT À ATTÉNUER CET ÉCART.

EN CE QUI CONCERNE LE CHOIX DE NESTJS AVEC TYPEORM POUR LA PARTIE BACK-END DE SHAREEVENT, CES TECHNOLOGIES OFFRENT PLUSIEURS AVANTAGES NOTABLES. NESTJS, EN TANT QUE FRAMEWORK BACK-END BASÉ SUR NODE.JS, FACILITE LE DÉVELOPPEMENT D'APPLICATIONS SERVEUR ROBUSTES ET ÉVOLUTIVES. SA STRUCTURE BIEN DÉFINIE, BASÉE SUR L'ARCHITECTURE ORIENTÉE VERS LES SERVICES (SOA) ET LA PROGRAMMATION ORIENTÉE OBJET (POO), OFFRE UNE BASE SOLIDE POUR LA GESTION DES ROUTES, L'INJECTION DE DÉPENDANCES ET LA GESTION DES REQUÊTES HTTP. DE PLUS, NESTJS BÉNÉFICIE D'UNE COMMUNAUTÉ ACTIVE ET DE NOMBREUSES RESSOURCES DISPONIBLES, CE QUI FACILITE L'APPRENTISSAGE ET LE DÉVELOPPEMENT.

TYPEORM, EN TANT QU'ORM COMPATIBLE AVEC NESTJS, SIMPLIFIE CONSIDÉRABLEMENT L'INTERACTION AVEC LA BASE DE DONNÉES EN UTILISANT DES OBJETS ET DES REQUÊTES EN LANGAGE TYPESCRIPT. CELA PERMET AUX DÉVELOPPEURS DE MODÉLISER ET DE MANIPULER LES DONNÉES DE MANIÈRE INTUITIVE, EN ÉVITANT LA NÉCESSITÉ D'ÉCRIRE DES REQUÊTES SQL MANUELLEMENT. TYPEORM OFFRE ÉGALEMENT DES FONCTIONNALITÉS TELLES QUE LA MIGRATION DE BASE DE DONNÉES ET LA GESTION DES RELATIONS ENTRE LES ENTITÉS, CE QUI FACILITE LA GESTION DE LA PERSISTANCE DES DONNÉES.

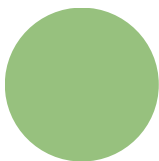
CEPENDANT, LE CHOIX DE NESTJS ET TYPEORM PEUT PRÉSENTER UNE COURBE D'APPRENTISSAGE INITIALE PLUS ÉLEVÉE, EN PARTICULIER POUR LES DÉVELOPPEURS MOINS FAMILIERS AVEC TYPESCRIPT OU LES FRAMEWORKS BACK-END BASÉS SUR NODE.JS. LA CONFIGURATION INITIALE ET LA COMPRÉHENSION DES CONCEPTS SPÉCIFIQUES À CES TECHNOLOGIES PEUVENT NÉCESSITER UN INVESTISSEMENT SUPPLÉMENTAIRE EN TERMES DE TEMPS ET DE RESSOURCES. CEPENDANT, UNE FOIS CETTE PHASE D'APPRENTISSAGE TERMINÉE, NESTJS ET TYPEORM OFFRENT UNE BASE SOLIDE ET COHÉRENTE POUR DÉVELOPPER ET MAINTENIR L'APPLICATION SHAREEVENT.

CONCEPTION DE L'APPLICATION

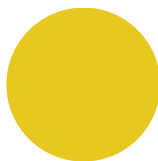
3.1 SPÉCIFICATIONS FONCTIONNELLES ET NON-FONCTIONNELLES :

LA PHASE DE CONCEPTION DE L'APPLICATION SHAREEVENT A DÉBUTÉ PAR L'IDENTIFICATION DES SPÉCIFICATIONS FONCTIONNELLES ET NON-FONCTIONNELLES. LES SPÉCIFICATIONS FONCTIONNELLES DÉCRIVENT LES FONCTIONNALITÉS ATTENDUES DE L'APPLICATION, TELLES QUE LA CRÉATION D'ÉVÉNEMENTS, L'INVITATION DE PARTICIPANTS, LE PARTAGE DE PHOTOS, ETC. LES SPÉCIFICATIONS NON-FONCTIONNELLES COUVRENT DES ASPECTS TELS QUE LA CONVIVIALITÉ, LES PERFORMANCES, LA SÉCURITÉ ET LA COMPATIBILITÉ MULTIPLATEFORME.

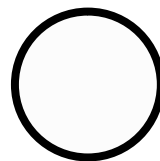
DE PLUS, L'ASPECT GRAPHIQUE A ÉTÉ RÉFLIÉ AVEC UNE MISE EN PLACE D'UNE CHARTE GRAPHIQUE C'EST À DIRE UN CODE COULEUR UNE FONT ET UNE IDÉE DE DESIGN AINSI QU'UN LOGO:



#97C17E



#E6C821



#FCFCFC

Typographie

CINZEL

OPEN SANS

3.2 ARCHITECTURE DE L'APPLICATION :

L'ARCHITECTURE DE SHAREEVENT A ÉTÉ CONÇUE POUR ASSURER LA MODULARITÉ, LA SCALABILITÉ ET LA MAINTENABILITÉ DE L'APPLICATION. UNE APPROCHE BASÉE SUR LES SERVICES A ÉTÉ ADOPTÉE, AVEC UNE SÉPARATION CLAIRE ENTRE LE FRONT-END MOBILE, LE BACK-END ET LA BASE DE DONNÉES. CETTE ARCHITECTURE FACILITE L'AJOUT DE NOUVELLES FONCTIONNALITÉS, LE TEST ET LE DÉPLOIEMENT DE L'APPLICATION.

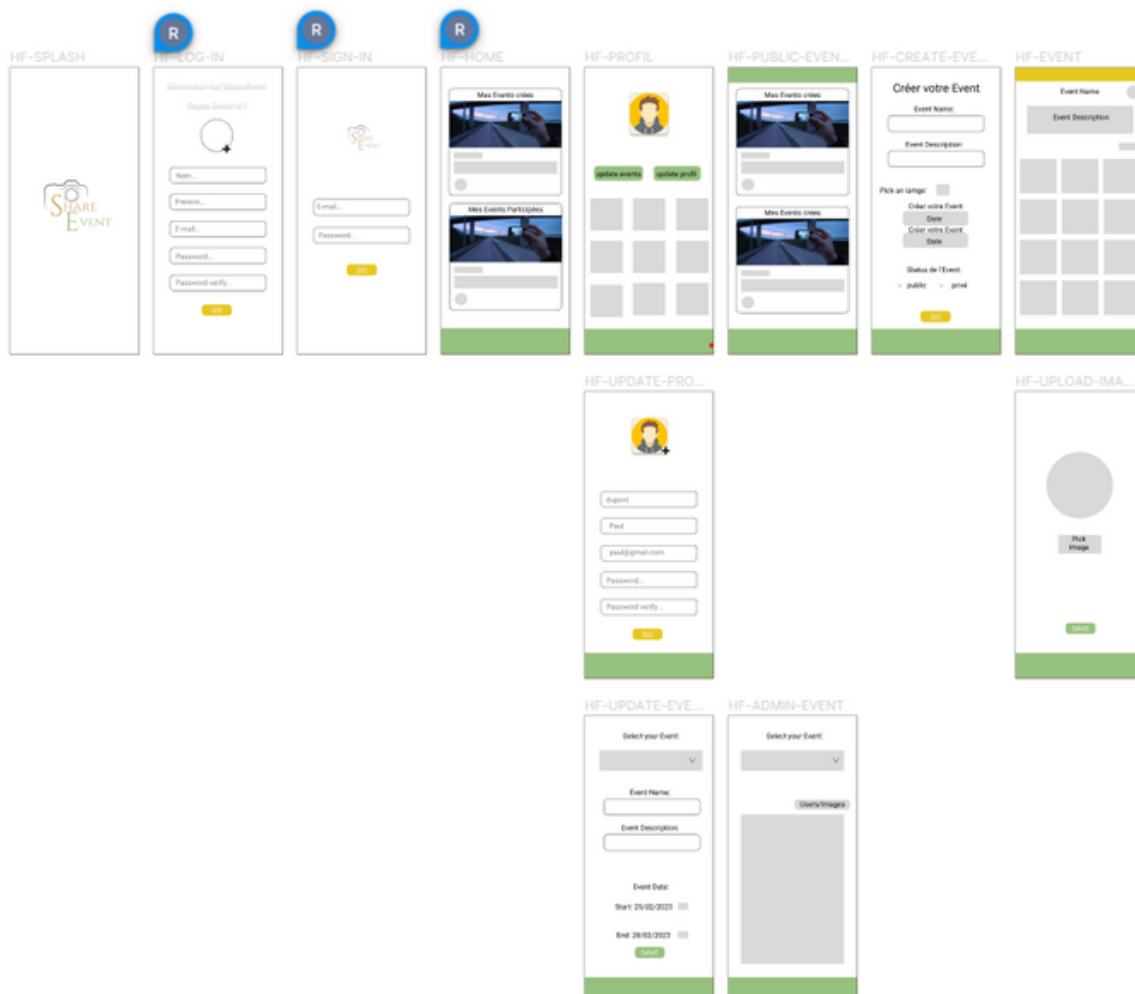
3.3 CONCEPTION DE L'INTERFACE UTILISATEUR AVEC FIGMA :

LA CONCEPTION DE L'INTERFACE UTILISATEUR DE SHAREEVENT A ÉTÉ RÉALISÉE EN UTILISANT L'OUTIL DE CONCEPTION FIGMA. À L'AIDE DE FIGMA, DES MAQUETTES INTERACTIVES ONT ÉTÉ CRÉÉES POUR REPRÉSENTER LES DIFFÉRENTES VUES DE L'APPLICATION. CELA A PERMIS DE VISUALISER L'APPARENCE ET LE FLUX DE L'APPLICATION AVANT MÊME LE DÉVELOPPEMENT.

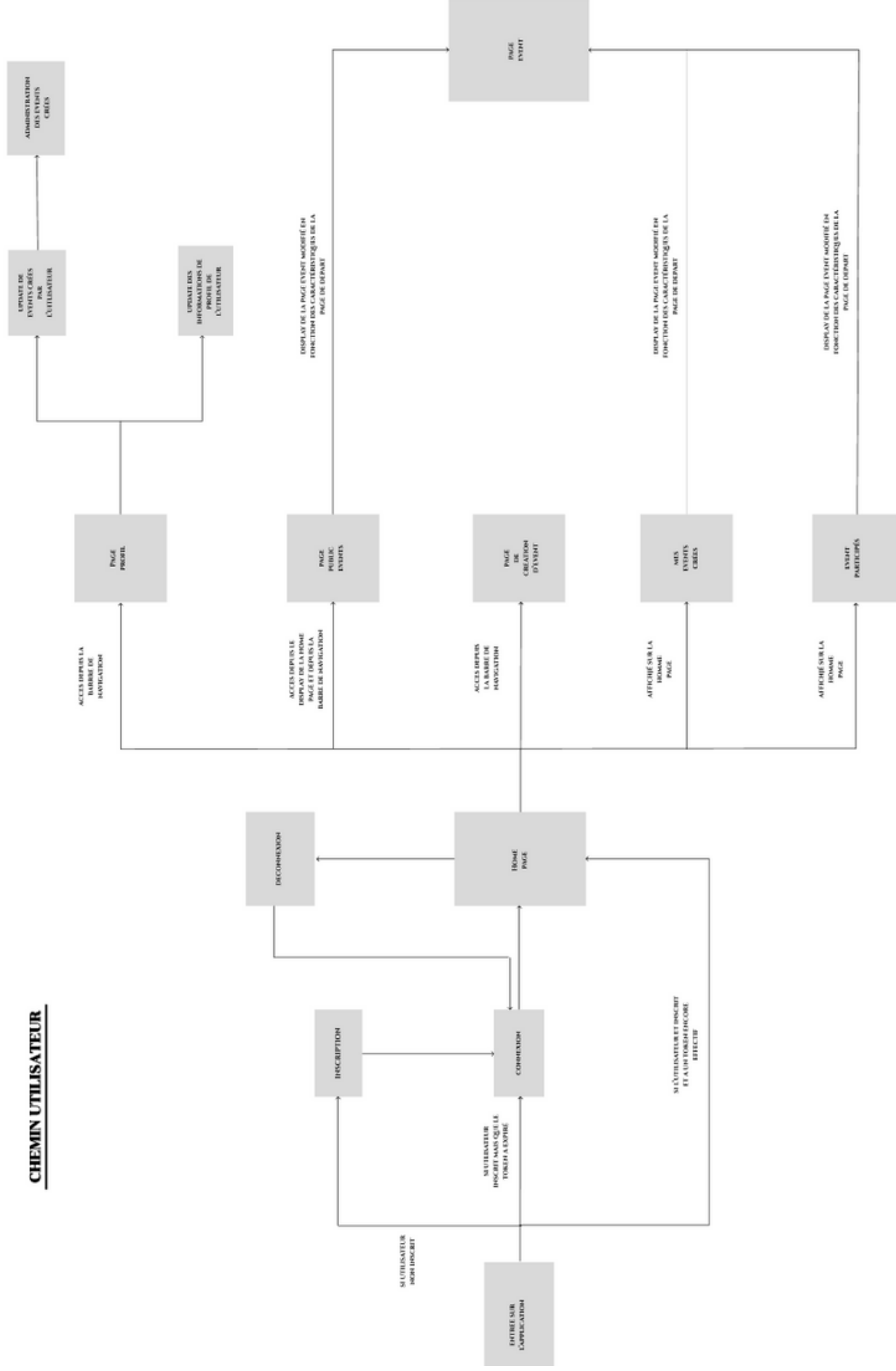
L'OBJECTIF PRINCIPAL ÉTAIT DE CRÉER UNE EXPÉRIENCE UTILISATEUR FLUIDE ET INTUITIVE. LES ÉLÉMENTS D'INTERFACE TELS QUE LES BOUTONS, LES FORMULAIRES, LES LISTES ET LES CARTES ONT ÉTÉ SOIGNEUSEMENT CONÇUS POUR ÊTRE COHÉRENTS ET FACILES À COMPRENDRE. LES COULEURS, LES TYPOGRAPHIES ET LES ICÔNES ONT ÉTÉ CHOISIS POUR REFLÉTER L'IDENTITÉ VISUELLE DE SHAREEVENT ET POUR ASSURER UNE EXPÉRIENCE ESTHÉTIQUEMENT AGRÉABLE.

EN UTILISANT FIGMA, LES PROTOTYPES INTERACTIFS ONT ÉTÉ PARTAGÉS AVEC L'ÉQUIPE DE DÉVELOPPEMENT ET LES PARTIES PRENANTES POUR OBTENIR DES RETOURS ET DES AMÉLIORATIONS. LES ITÉRATIONS SUCCESSIVES ONT PERMIS D'AJUSTER ET D'AFFINER L'INTERFACE UTILISATEUR EN FONCTION DES COMMENTAIRES REÇUS.

MAQUETTE HAUTE FIDELITE



CHEMIN UTILISATEUR

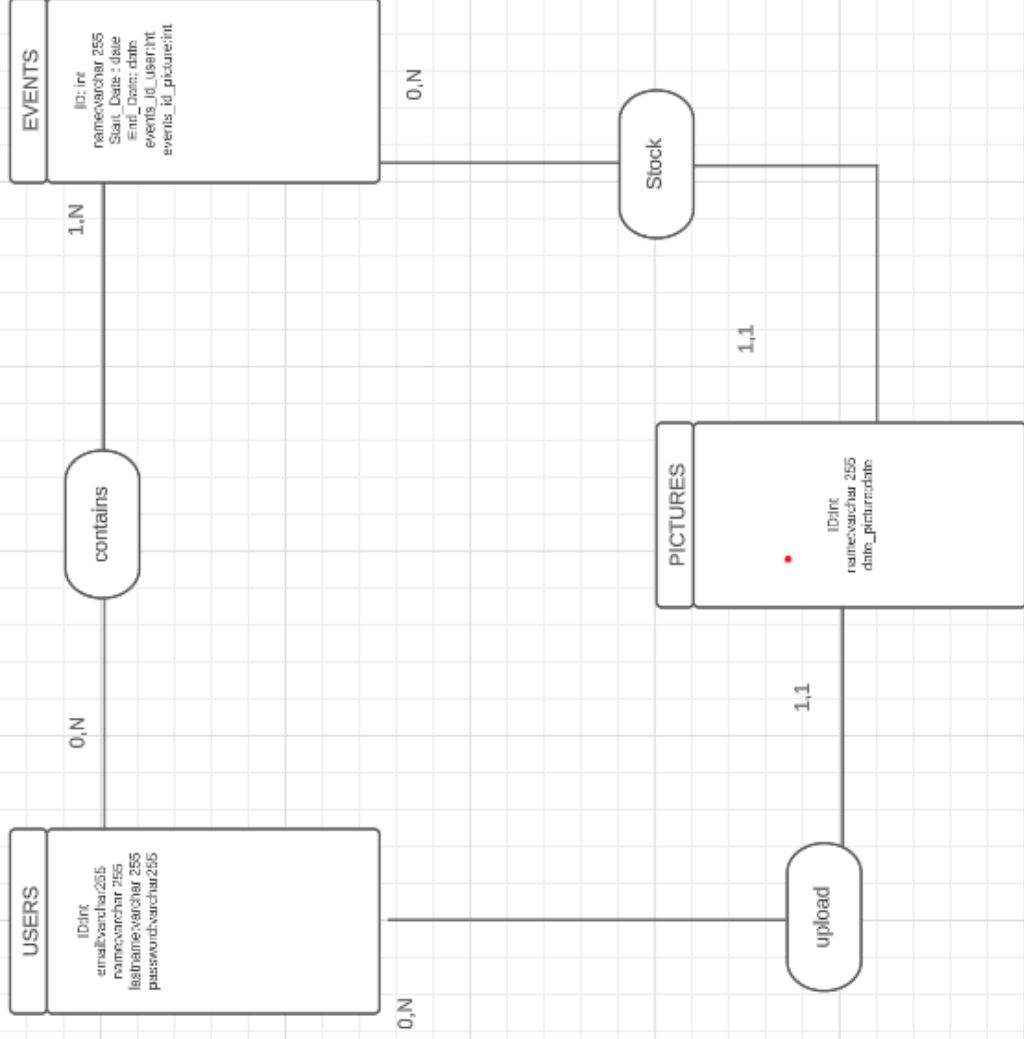


3.4 CONCEPTION DE LA BASE DE DONNÉES AVEC LUCIDCHART ET LA MÉTHODE MERISE :

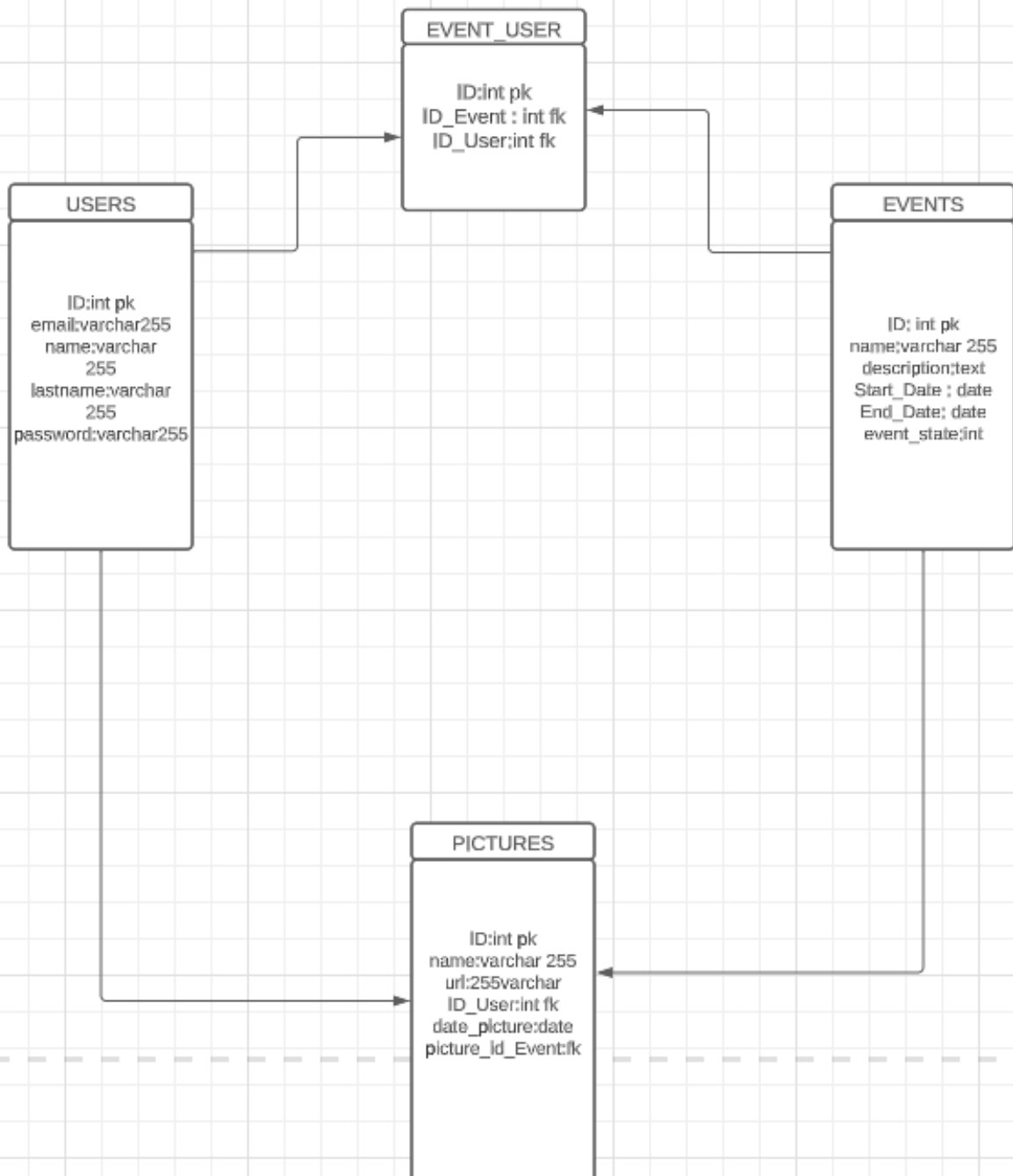
POUR CONCEVOIR LA BASE DE DONNÉES DE SHAREEVENT, L'OUTIL DE MODÉLISATION LUCIDCHART A ÉTÉ UTILISÉ EN SUIVANT LA MÉTHODE MERISE. LA MÉTHODE MERISE OFFRE UNE APPROCHE STRUCTURÉE POUR LA MODÉLISATION CONCEPTUELLE, LOGIQUE ET PHYSIQUE DES DONNÉES. LES ENTITÉS PRINCIPALES TELLES QUE LES UTILISATEURS, LES ÉVÉNEMENTS ET LES PHOTOS ONT ÉTÉ IDENTIFIÉES, AINSI QUE LES RELATIONS ET LES ATTRIBUTS ASSOCIÉS. LE SCHÉMA DE BASE DE DONNÉES RÉSULTANT A ÉTÉ UTILISÉ COMME GUIDE POUR L'IMPLÉMENTATION AVEC TYPEORM.

LA CONCEPTION GLOBALE DE L'APPLICATION SHAREEVENT, COMPRENANT L'ARCHITECTURE, L'INTERFACE UTILISATEUR ET LA BASE DE DONNÉES, A ÉTÉ RÉALISÉE EN VEILLANT À RÉPONDRE AUX BESOINS FONCTIONNELS ET NON-FONCTIONNELS IDENTIFIÉS. CETTE PHASE DE CONCEPTION SOLIDE, INCLUANT LA MAQUETTE ET L'EXPÉRIENCE UTILISATEUR, A JETÉ LES BASES D'UN DÉVELOPPEMENT EFFICACE ET COHÉRENT DE L'APPLICATION.

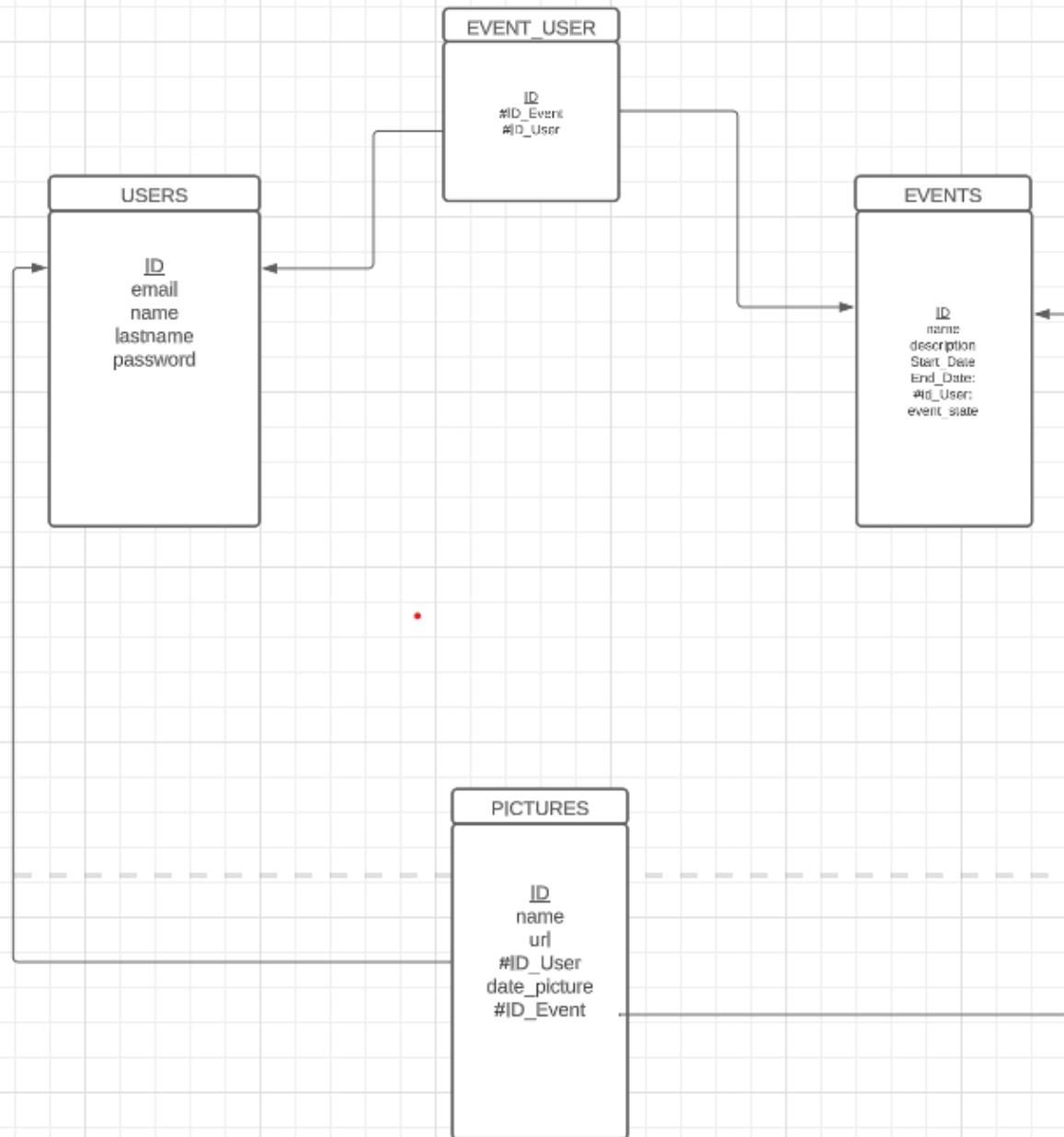
MCD



MLD



MPD



DÉVELOPPEMENT DE LA PARTIE FRONT-END

4.1 PRÉSENTATION DE REACT NATIVE :

POUR LE DÉVELOPPEMENT DE LA PARTIE FRONT-END DE SHAREEVENT, NOUS AVONS UTILISÉ REACT NATIVE, UN FRAMEWORK JAVASCRIPT LARGEMENT UTILISÉ POUR LA CRÉATION D'APPLICATIONS MOBILES MULTIPLATEFORMES. REACT NATIVE PERMET DE DÉVELOPPER DES APPLICATIONS POUR IOS ET ANDROID EN UTILISANT UNE BASE DE CODE COMMUNE, CE QUI PERMET D'ÉCONOMISER DU TEMPS ET DES RESSOURCES LORS DU DÉVELOPPEMENT.

L'AVANTAGE DE REACT NATIVE RÉSIDE DANS SA CAPACITÉ À CRÉER DES INTERFACES UTILISATEUR RÉACTIVES ET PERFORMANTES EN UTILISANT DES COMPOSANTS PRÉFABRIQUÉS RÉUTILISABLES. IL UTILISE UNE ARCHITECTURE BASÉE SUR DES COMPOSANTS, CE QUI FACILITE LA MODULARITÉ ET LA RÉUTILISABILITÉ DU CODE. DE PLUS, REACT NATIVE PERMET D'ACCÉDER AUX FONCTIONNALITÉS NATIVES DES APPAREILS MOBILES GRÂCE À DES BIBLIOTHÈQUES SPÉCIFIQUES.

NOUS AVONS UTILISÉ EXPO QUI EST UN FRAMEWORK OPEN SOURCE POUR LE DÉVELOPPEMENT D'APPLICATIONS MOBILES, QUI PERMET DE CRÉER DES APPLICATIONS POUR IOS, ANDROID ET LE WEB EN UTILISANT LE LANGAGE JAVASCRIPT ET LA BIBLIOTHÈQUE REACT. EXPO FOURNIT DES OUTILS ET DES SERVICES POUR FACILITER LE DÉVELOPPEMENT, TELS QUE LE CHARGEMENT SANS FIL DES APPLICATIONS SUR LES APPAREILS, L'ACCÈS AUX FONCTIONNALITÉS DU TÉLÉPHONE (APPAREIL PHOTO, GÉOLOCALISATION, ETC.), ET LA PRISE EN CHARGE DE DIVERSES BIBLIOTHÈQUES ET API.



4.2 IMPLÉMENTATION DES FONCTIONNALITÉS DE L'APPLICATION :

DANS SHAREEVENT, PLUSIEURS FONCTIONNALITÉS CLÉS ONT ÉTÉ IMPLÉMENTÉES DANS LA PARTIE FRONT-END À L'AIDE DE REACT NATIVE. CELA INCLUT :

CRÉATION D'ÉVÉNEMENTS : LES UTILISATEURS PEUVENT CRÉER DE NOUVEAUX ÉVÉNEMENTS EN FOURNISSANT DES DÉTAILS TELS QUE LE TITRE, LA DATE, L'EMPLACEMENT, ETC. DES FORMULAIRES ONT ÉTÉ MIS EN PLACE POUR COLLECTER CES INFORMATIONS ET LES VALIDER AVANT LEUR ENREGISTREMENT.

AFFICHAGE DES ÉVÉNEMENTS : UNE FOIS LES ÉVÉNEMENTS CRÉÉS, ILS SONT AFFICHÉS DANS UNE LISTE POUR QUE LES UTILISATEURS PUISSENT LES CONSULTER. CHAQUE ÉVÉNEMENT EST PRÉSENTÉ AVEC SES DÉTAILS PERTINENTS, TELS QUE LE TITRE, LA DATE ET L'EMPLACEMENT.

INVITATION DES PARTICIPANTS : LES UTILISATEURS PEUVENT INVITER D'AUTRES PARTICIPANTS À LEURS ÉVÉNEMENTS EN SAISSANT LEURS COORDONNÉES. DES MÉCANISMES DE RECHERCHE ET DE SÉLECTION DE CONTACTS ONT ÉTÉ INTÉGRÉS POUR FACILITER CE PROCESSUS.

PARTAGE DE PHOTOS : SHAREEVENT PERMET AUX UTILISATEURS DE PARTAGER DES PHOTOS LIÉES À UN ÉVÉNEMENT SPÉCIFIQUE. ILS PEUVENT PRENDRE DES PHOTOS DIRECTEMENT DEPUIS L'APPLICATION OU SÉLECTIONNER DES IMAGES EXISTANTES DANS LEUR GALERIE. CES PHOTOS SONT ENSUITE ASSOCIÉES À L'ÉVÉNEMENT CORRESPONDANT.

```
const {width} = Dimensions.get('window');
const SPACING = 5;
const ITEM_LENGTH = width * 1; // Item is a square. Therefore, its height and width are of the same length.
const EMPTY_ITEM_LENGTH = (width - ITEM_LENGTH) / 2;
const BORDER_RADIUS = 20;
const CURRENT_ITEM_TRANSLATE_Y = 20;

// interface ImageCarouselProps {
//   data: ImageCarouselItem[];
// }

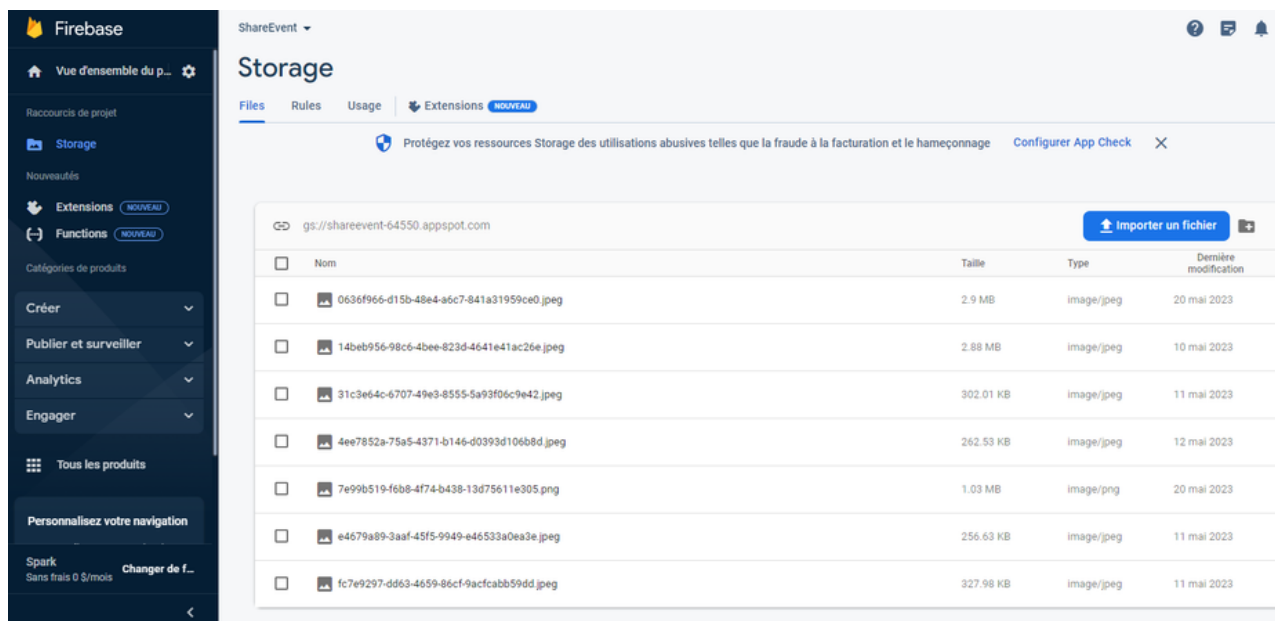
export const CarouselEvent=()=> {

  const[data,setData]=useState<any[]>([]);
  useEffect(() => {
    async function GetAllEvent() {
      try {
        const response: any = await axios.get("http://10.10.22.116:3000/events");
        console.log(response.data);
        setData(response.data);
      }
      catch (error) {
        console.error(error);
      }
    }
    GetAllEvent(),
    []);

  const scrollX = useRef(new Animated.Value(0)).current;
```

FIREBASE OFFRE UN SERVICE DE STOCKAGE D'OBJETS APPELÉ FIREBASE STORAGE, QUI PERMET DE STOCKER ET DE GÉRER DES FICHIERS, Y COMPRIS DES IMAGES, DE MANIÈRE SIMPLE ET SÉCURISÉE. AVEC FIREBASE STORAGE, LES DÉVELOPPEURS PEUVENT FACILEMENT TÉLÉCHARGER, RÉCUPÉRER ET SUPPRIMER DES IMAGES DANS LE CLOUD, EN UTILISANT UNE API SIMPLE ET DES BIBLIOTHÈQUES ADAPTÉES À DIFFÉRENTS LANGAGES DE PROGRAMMATION.

LORSQU'UNE IMAGE EST TÉLÉCHARGÉE, FIREBASE STORAGE GÉNÈRE AUTOMATIQUEMENT UNE URL UNIQUE POUR Y ACCÉDER. CETTE URL PEUT ÊTRE UTILISÉE POUR AFFICHER L'IMAGE DANS UNE APPLICATION OU UN SITE WEB. FIREBASE STORAGE OFFRE ÉGALEMENT DES FONCTIONNALITÉS TELLES QUE LE CONTRÔLE DES AUTORISATIONS D'ACCÈS AUX FICHIERS, LA GESTION DES VERSIONS, LE SUIVI DES MÉTADONNÉES ET LA MISE À L'ÉCHELLE AUTOMATIQUE POUR S'ADAPTER À LA DEMANDE DE STOCKAGE. GRÂCE À CETTE INTÉGRATION TRANSPARENTE AVEC LES AUTRES SERVICES FIREBASE, LES DÉVELOPPEURS PEUVENT CONSTRUIRE DES APPLICATIONS DYNAMIQUES ET INTERACTIVES QUI GÈRENT EFFICACEMENT LE STOCKAGE ET LE PARTAGE D'IMAGES.



The screenshot displays the Firebase Storage console interface. On the left is a dark sidebar with the Firebase logo and navigation options like 'Vue d'ensemble du p...', 'Storage', 'Extensions', and 'Fonctions'. The main area is titled 'Storage' and shows a list of files under the 'Files' tab. A warning banner at the top of the main area states: 'Protégez vos ressources Storage des utilisations abusives telles que la fraude à la facturation et le hameçonnage'. Below this, there's a link to 'Configurer App Check'. The file list table has columns for 'Nom', 'Taille', 'Type', and 'Dernière modification'. It contains 8 rows of files, all with image file extensions (.jpeg or .png).

Nom	Taille	Type	Dernière modification
0636f966-d15b-48e4-a6c7-841a31959ce0.jpeg	2.9 MB	image/jpeg	20 mai 2023
14beb956-98c6-4bee-823d-4641e41ac26e.jpeg	2.88 MB	image/jpeg	10 mai 2023
31c3e64c-6707-49e3-8555-5a93f06c9e42.jpeg	302.01 KB	image/jpeg	11 mai 2023
4ee7852a-75a5-4371-b146-d0393d106b8d.jpeg	262.53 KB	image/jpeg	12 mai 2023
7e99b519-f6b8-4f74-b438-13d75611e305.png	1.03 MB	image/png	20 mai 2023
e4679a89-3aaf-45f5-9949-e46533a0ea3e.jpeg	256.63 KB	image/jpeg	11 mai 2023
fc7e9297-dd63-4659-86cf-9acfcabb59dd.jpeg	327.98 KB	image/jpeg	11 mai 2023

```

const saveImage2 = async (uri: string, eventId: number | null) => {
  try {
    const response = await fetch(imageUri);
    const blob = await response.blob();
    const filename = imageUri.substring(imageUri.lastIndexOf("/") + 1);
    var ref = firebase.storage().ref().child(filename).put(blob);

    try {
      ref.then(async (snapshot) => {
        const url = await snapshot.ref.getDownloadURL();
        console.log(url);
        console.log(userId);

        // Conditionally update the request data based on the uploadPurpose prop
        const requestData =
          uploadPurpose === "event"
            ? {
                name: filename,
                url: url,
                userId: userId,
                eventId: eventId,
              }
            : {
                name: filename,
                url: url,
                userId: userId,
              };

        const response = await axios.post(
          `http://192.168.1.27:3000/pictures/register`,
          requestData
        );
        console.log("Response from server:", response); // Log the server response
        if (onImageUploaded) {
          onImageUploaded(url);
        }
      });
    }
  }
};

```

4.3 GESTION DE L'AUTHENTIFICATION ET DES AUTORISATIONS :

DANS SHAREEVENT, LA GESTION DE L'AUTHENTIFICATION DES UTILISATEURS ET DES AUTORISATIONS EST ÉGALEMENT PRISE EN COMPTE DANS LA PARTIE FRONT-END. CELA GARANTIT QUE SEULS LES UTILISATEURS AUTHENTIFIÉS ET AUTORISÉS PEUVENT ACCÉDER AUX FONCTIONNALITÉS DE L'APPLICATION.

L'AUTHENTIFICATION DES UTILISATEURS EST RÉALISÉE EN DEMANDANT DES INFORMATIONS D'IDENTIFICATION TELLES QU'UN NOM D'UTILISATEUR ET UN MOT DE PASSE. DES MÉCANISMES DE SÉCURISATION, TELS QUE LE HACHAGE DES MOTS DE PASSE, SONT UTILISÉS POUR PROTÉGER LES INFORMATIONS CONFIDENTIELLES DES UTILISATEURS.

LES AUTORISATIONS SONT MISES EN PLACE POUR CONTRÔLER L'ACCÈS AUX DIFFÉRENTES FONCTIONNALITÉS DE L'APPLICATION. PAR EXEMPLE, SEULS LES UTILISATEURS QUI ONT CRÉÉ UN ÉVÉNEMENT PEUVENT MODIFIER SES DÉTAILS OU SUPPRIMER L'ÉVÉNEMENT. LES AUTORISATIONS SONT GÉRÉES EN FONCTION DU RÔLE ET DES PRIVILÈGES DE CHAQUE UTILISATEUR.

EN RÉSUMÉ, LA PARTIE FRONT-END DE SHAREEVENT A ÉTÉ DÉVELOPPÉE EN UTILISANT REACT NATIVE, OFFRANT AINSI UNE INTERFACE UTILISATEUR RÉACTIVE ET PERFORMANTE. LES FONCTIONNALITÉS CLÉS DE L'APPLICATION, TELLES QUE LA CRÉATION D'ÉVÉNEMENTS, L'AFFICHAGE DES ÉVÉNEMENTS, L'INVITATION

```
const [isLoggedIn, setIsLoggedIn] = useState(false);
const handleLogin = async () => {
  setError(null);
  setIsLoading(true);

  if (!validateEmail(email)) {
    setError({ message: 'Invalid email format', statusCode: 0 });
    setIsLoading(false);
    return;
  }

  try {
    const response = await axios.post(`http://192.168.1.27:3000/auth/login`, { email, password });
    await SecureStore.setItemAsync('access_token', response.data.access_token);
    await SecureStore.setItemAsync('refresh_token', response.data.refresh_token);

    setIsLoading(false);
    setIsLoggedIn(true);
    setError({ message: 'Login successful', statusCode: 200 });

    // Use navigation.dispatch to reset the navigation stack and navigate to the Home screen
    navigation.dispatch(
      CommonActions.reset({
        index: 0,
        routes: [{ name: 'EventsStack' }],
      })
    );
  } catch (e) {
    setError({ message: 'Invalid email or password', statusCode: 0 });
    setIsLoading(false);
  }
}
```

4.4 COMMUNICATION FRONT/BACK :AXIOS

AXIOS EST UNE BIBLIOTHÈQUE JAVASCRIPT POPULAIRE UTILISÉE POUR EFFECTUER DES REQUÊTES HTTP DEPUIS UN NAVIGATEUR OU DEPUIS UN SERVEUR UTILISANT NODE.JS. ELLE SIMPLIFIE ET FACILITE LES APPELS RÉSEAU EN FOURNISSANT UNE INTERFACE SIMPLE ET INTUITIVE POUR INTERAGIR AVEC DES API ET RÉCUPÉRER DES DONNÉES A PARTIR DE SERVEURS DISTANTS. VOICI POURQUOI AXIOS EST APPRÉCIÉ :

SYNTAXE ÉLÉGANTE : AXIOS UTILISE UNE SYNTAXE CLAIRE ET EXPRESSIVE POUR EFFECTUER DES REQUÊTES HTTP. IL PERMET D'EFFECTUER DES REQUÊTES AVEC DIFFÉRENTES MÉTHODES TELLES QUE GET, POST, PUT, DELETE, ETC. L'UTILISATION DE PROMÈSSES PERMET DE GÉRER FACILEMENT LES RÉPONSES ASYNCHRONES.

COMPATIBILITÉ DES NAVIGATEURS : AXIOS EST COMPATIBLE AVEC LA PLUPART DES NAVIGATEURS MODERNES, Y COMPRIS LES ANCIENNES VERSIONS D'INTERNET EXPLORER. IL UTILISE LES API XMLHttpRequest ou LA MÉTHODE FETCH DU NAVIGATEUR POUR EFFECTUER LES REQUÊTES, GARANTISSANT UNE COMPATIBILITÉ ÉTENDUE.

INTERCEPTION DES REQUÊTES ET DES RÉPONSES : AXIOS OFFRE LA POSSIBILITÉ D'INTERCEPTER LES REQUÊTES ET LES RÉPONSES AVANT QU'ELLES NE SOIENT ENVOYÉES OU TRAITÉES. CELA PERMET DE MANIPULER LES DONNÉES, D'AJOUTER DES EN-TÊTES PERSONNALISÉES, DE GÉRER LES ERREURS DE MANIÈRE CENTRALISÉE, OU D'AUTHENTIFIER LES REQUÊTES AVEC DES JETONS D'ACCÈS.

GESTION AVANCÉE DES ERREURS : AXIOS FACILITE LA GESTION DES ERREURS LIÉES AUX REQUÊTES HTTP. IL CAPTURE LES ERREURS RÉSEAU, LES ERREURS DE CONNEXION, LES ERREURS HTTP ET PERMET DE LES TRAITER DE MANIÈRE COHÉRENTE. DE PLUS, IL FOURNIT DES MÉCANISMES POUR DÉFINIR DES INTERCEPTEURS D'ERREURS GLOBAUX ET SPÉCIFIQUES À UNE REQUÊTE, AFIN DE GÉRER LES ÉCHECS DE MANIÈRE PLUS EFFICACE.

FONCTIONNALITÉS SUPPLÉMENTAIRES : AXIOS OFFRE DES FONCTIONNALITÉS SUPPLÉMENTAIRES TELLES QUE L'ANNULATION DES REQUÊTES EN COURS, L'ENVOI DE DONNÉES SOUS FORME DE FORMULAIRE, LA GESTION DES COOKIES, LE TÉLÉCHARGEMENT DE FICHIERS, ET BIEN PLUS ENCORE. CES FONCTIONNALITÉS ÉTENDUES FONT D'AXIOS UNE BIBLIOTHÈQUE POLYVALENTE POUR LA GESTION DES REQUÊTES HTTP.

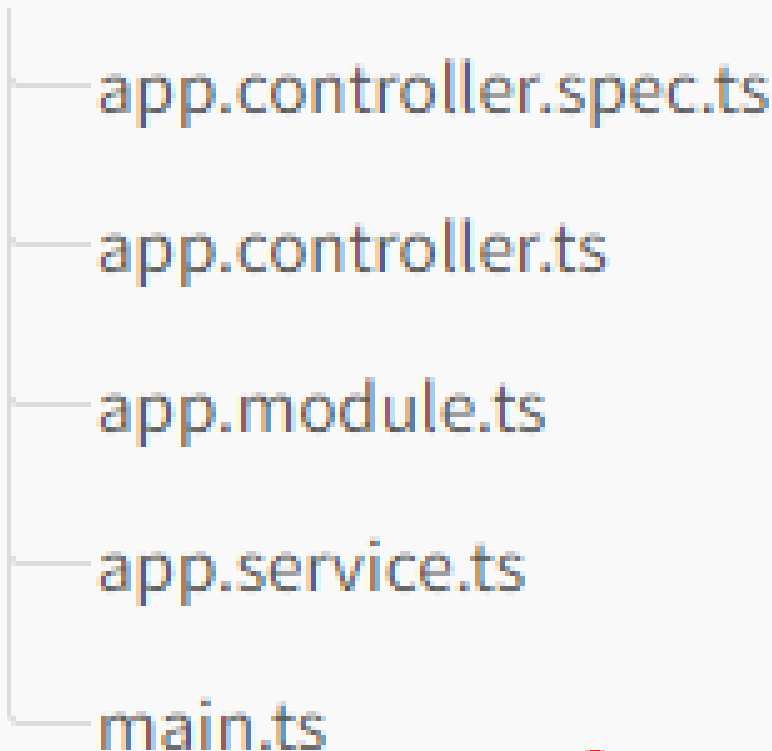
DÉVELOPPEMENT DE LA PARTIE BACK-END

5.1 INTRODUCTION A NEST JS

POUR LE DÉVELOPPEMENT DE LA PARTIE BACK-END DE SHAREEVENT, NOUS AVONS UTILISÉ LE FRAMEWORK NESTJS. NESTJS EST UN FRAMEWORK NODE.JS BASÉ SUR TYPESCRIPT QUI OFFRE UNE ARCHITECTURE SOLIDE ET MODULAIRE POUR LA CRÉATION D'APPLICATIONS BACK-END ÉVOLUTIVES ET MAINTENABLES.

NESTJS SUIT LE MODÈLE ARCHITECTURAL MVC (MODÈLE-VUE-CONTRÔLEUR) ET ENCOURAGE L'UTILISATION DE BONNES PRATIQUES TELLES QUE L'INJECTION DE DÉPENDANCES, LA SÉPARATION DES PRÉOCCUPATIONS ET LA MODULARITÉ DU CODE. IL UTILISE EXPRESS.JS COMME COUCHE SOUS-JACENTE POUR LA GESTION DES REQUÊTES ET DES RÉPONSES HTTP.

src



```
graph LR; src --> app_controller_spec_ts[app.controller.spec.ts]; src --> app_controller_ts[app.controller.ts]; src --> app_module_ts[app.module.ts]; src --> app_service_ts[app.service.ts]; src --> main_ts[main.ts];
```

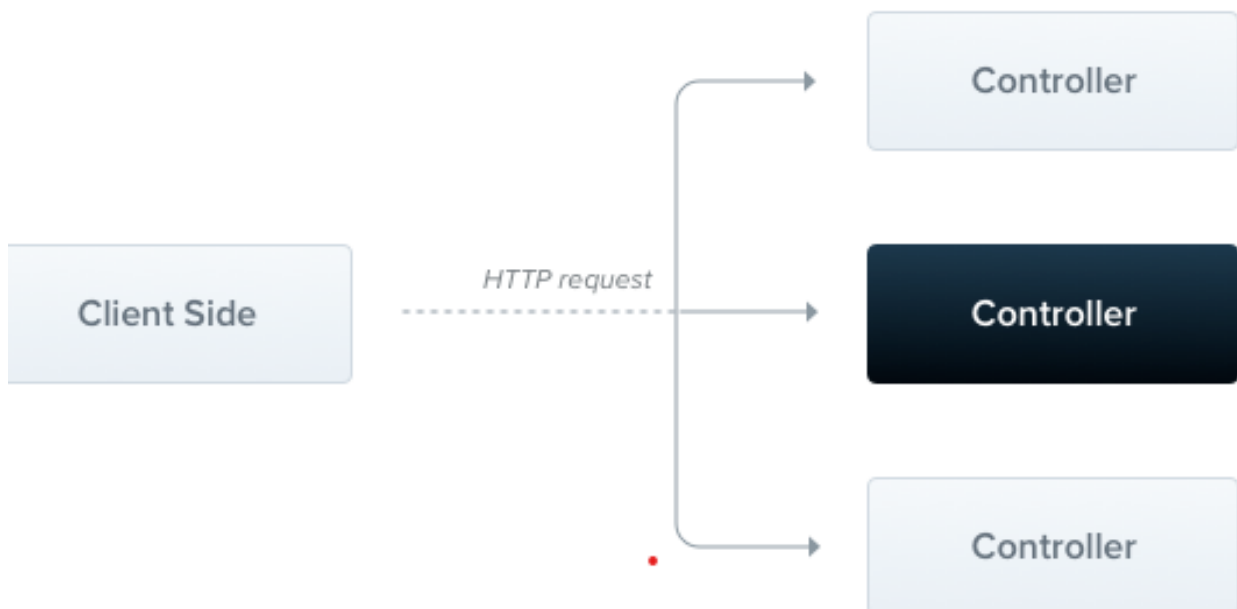
The diagram shows a file structure for the 'src' directory. A vertical line on the left represents the directory, with horizontal lines branching off to the right to list the files: 'app.controller.spec.ts', 'app.controller.ts', 'app.module.ts', 'app.service.ts', and 'main.ts'.

5.2 MODÉLISATION DE LA BASE DE DONNÉES AVEC TYPEORM :

POUR LA MODÉLISATION DE LA BASE DE DONNÉES DE SHAREEVENT, NOUS AVONS UTILISÉ TYPEORM, UN ORM (OBJECT-RELATIONAL MAPPING) COMPATIBLE AVEC DE NOMBREUX SYSTÈMES DE GESTION DE BASES DE DONNÉES RELATIONNELLES TELS QUE MYSQL, POSTGRESQL, SQLITE, ETC.

TYPEORM PERMET DE DÉFINIR DES ENTITÉS EN UTILISANT DES CLASSES TYPESCRIPT/JAVASCRIPT QUI CORRESPONDENT AUX TABLES DE LA BASE DE DONNÉES. LES PROPRIÉTÉS DE CES ENTITÉS SONT MAPPÉES AUX COLONNES DE LA BASE DE DONNÉES, ET LES RELATIONS ENTRE LES ENTITÉS SONT ÉGALEMENT DÉFINIES.

GRÂCE À TYPEORM, NOUS AVONS PU CRÉER DES MODÈLES DE DONNÉES COHÉRENTS POUR LES ENTITÉS TELLES QUE LES ÉVÉNEMENTS, LES UTILISATEURS ET LES PHOTOS. LES RELATIONS ENTRE CES ENTITÉS, TELLES QUE LES RELATIONS UN-À-PLUSIEURS OU PLUSIEURS-À-PLUSIEURS, ONT ÉTÉ DÉFINIES POUR REFLÉTER LA STRUCTURE DE LA BASE DE DONNÉES.



5.3 IMPLÉMENTATION DES FONCTIONNALITÉS DU SERVEUR :

DANS LE CADRE DU DÉVELOPPEMENT DE SHAREEVENT, PLUSIEURS FONCTIONNALITÉS CLÉS ONT ÉTÉ IMPLÉMENTÉES DANS LA PARTIE BACK-END DU SERVEUR :

GESTION DES ÉVÉNEMENTS : DES ROUTES ET DES CONTRÔLEURS ONT ÉTÉ CRÉÉS POUR PERMETTRE LA MANIPULATION DES ÉVÉNEMENTS. CELA INCLUT LA CRÉATION D'UN NOUVEL ÉVÉNEMENT, LA RÉCUPÉRATION DE LA LISTE DES ÉVÉNEMENTS EXISTANTS, LA MISE À JOUR DES DÉTAILS D'UN ÉVÉNEMENT ET LA SUPPRESSION D'UN ÉVÉNEMENT. LES OPÉRATIONS CORRESPONDANTES ONT ÉTÉ IMPLÉMENTÉES EN UTILISANT LES MÉTHODES APPROPRIÉES DE TYPEORM POUR INTERAGIR AVEC LA BASE DE DONNÉES.

GESTION DES UTILISATEURS : DES FONCTIONNALITÉS D'INSCRIPTION, DE CONNEXION ET DE GESTION DES UTILISATEURS ONT ÉTÉ MISES EN PLACE. LES ROUTES ET LES CONTRÔLEURS CORRESPONDANTS ONT ÉTÉ CRÉÉS POUR GÉRER CES FONCTIONNALITÉS. PAR EXEMPLE, LORS DE L'INSCRIPTION D'UN NOUVEL UTILISATEUR, LES INFORMATIONS FOURNIES SONT VÉRIFIÉES, LE MOT DE PASSE EST HACHÉ AVANT D'ÊTRE STOCKÉ DANS LA BASE DE DONNÉES, ET UN JETON JWT (JSON WEB TOKEN) EST GÉNÉRÉ POUR L'AUTHENTIFICATION ULTÉRIEURE.

```
async login(GetUserDto: GetUserDto) {
  console.log(GetUserDto.email)
  const foundUser = await this.UsersService.findOneByEmail(GetUserDto.email);
  if (!foundUser) {
    return "l'email est incorrect";
  }
  const passwordMatch = await bcrypt.compare(GetUserDto.password, foundUser.password);
  if (!passwordMatch) {
    return 'l email ou le mot de passe est incorrect'
  }
  // const hashedRefreshToken = await bcrypt.hash(refreshToken, 10);
  const payload: UserInterface = await {
    id: foundUser.id,
    url: foundUser.url,
    email: foundUser.email,
    name: foundUser.name,
    lastname: foundUser.lastname,
    isAdmin: foundUser.isAdmin,
  }

  const accessToken = this.jwtService.sign(payload, { expiresIn: '30 days' });
  const refreshToken = this.jwtService.sign(payload, { expiresIn: '30 days' });

  return {
    access_token: accessToken,
    refresh_token: refreshToken,
  }
}
```

AUTHENTIFICATION ET AUTORISATION : DES STRATÉGIES D'AUTHENTIFICATION ONT ÉTÉ IMPLÉMENTÉES POUR SÉCURISER L'ACCÈS AUX FONCTIONNALITÉS PROTÉGÉES. LORSQU'UN UTILISATEUR SE CONNECTE, SES INFORMATIONS D'IDENTIFICATION SONT VÉRIFIÉES, ET S'IL EST VALIDE, UN JETON JWT EST GÉNÉRÉ ET RENVOYÉ AU CLIENT POUR UNE UTILISATION ULTÉRIEURE. LES ROUTES PROTÉGÉES VÉRIFIENT LA VALIDITÉ DU JETON ET ACCORDENT L'ACCÈS UNIQUEMENT AUX UTILISATEURS AUTHENTIFIÉS ET AUTORISÉS.

GESTION DES PHOTOS : DES FONCTIONNALITÉS DE TÉLÉCHARGEMENT ET DE STOCKAGE DES PHOTOS ONT ÉTÉ MISES EN PLACE. LES ROUTES ET LES CONTRÔLEURS CORRESPONDANTS ONT ÉTÉ CRÉÉS POUR GÉRER LE TÉLÉCHARGEMENT DES PHOTOS, LEUR ASSOCIATION AUX ÉVÉNEMENTS ET LEUR STOCKAGE. SELON LES BESOINS DE L'APPLICATION, DES BIBLIOTHÈQUES APPROPRIÉES (COMME MULTER POUR LE TÉLÉCHARGEMENT DES FICHIERS) ONT PU ÊTRE UTILISÉES POUR FACILITER CES FONCTIONNALITÉS.

DE PLUS, DES MÉCANISMES DE VALIDATION ET DE GESTION DES ERREURS ONT ÉTÉ MIS EN PLACE POUR GARANTIR LA COHÉRENCE DES DONNÉES ET LA ROBUSTESSE DE L'APPLICATION. DES VALIDATIONS ONT ÉTÉ EFFECTUÉES AU NIVEAU DU SERVEUR POUR S'ASSURER QUE LES DONNÉES REÇUES SONT CONFORMES AUX ATTENTES, ET DES RÉPONSES D'ERREUR APPROPRIÉES SONT RENVOYÉES EN CAS DE PROBLÈMES.

EN RÉSUMÉ, LA PARTIE BACK-END DU SERVEUR A ÉTÉ IMPLÉMENTÉE EN METTANT L'ACCENT SUR LES FONCTIONNALITÉS ESSENTIELLES DE SHAREEVENT TELLES QUE LA GESTION DES ÉVÉNEMENTS, DES UTILISATEURS, DE L'AUTHENTIFICATION ET DES AUTORISATIONS, AINSI QUE LA GESTION DES PHOTOS. CES FONCTIONNALITÉS ONT ÉTÉ DÉVELOPPÉES EN UTILISANT NESTJS ET TYPEORM, EN RESPECTANT LES BONNES PRATIQUES DE DÉVELOPPEMENT ET EN METTANT L'ACCENT SUR LA SÉCURITÉ ET LA FIABILITÉ. DES TESTS UNITAIRES ONT ÉTÉ RÉALISÉS POUR ASSURER LA QUALITÉ DU CODE ET LA STABILITÉ DE L'APPLICATION.

```
canActivate(context: ExecutionContext): boolean | Promise<boolean> | Observable<boolean> {
  const request = context.switchToHttp().getRequest();
  const token = request.headers.authorization; // assuming the token is sent in the "Authorization" header
  console.log(token);
  if (token == null || token == undefined) {
    return false;
  }

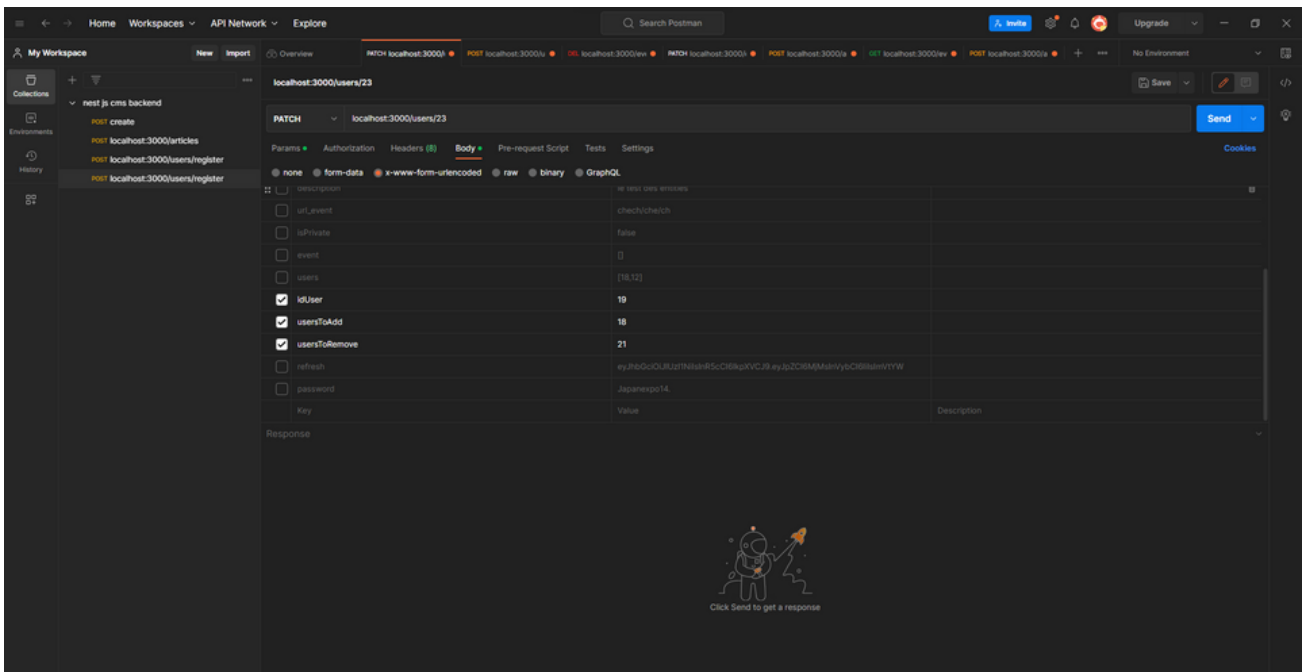
  try{

    const decodedToken = this.jwtService.decode(token) as { isAdmin: boolean };
    console.log(decodedToken.isAdmin)
    if(decodedToken.isAdmin == true){
      return true
    }
    else{
      return false
    }
  }
  catch{
    return false
  }
}
```

TEST DE VALIDATION

6.1 STRATÉGIE DE TESTS :

POUR GARANTIR LA QUALITÉ ET LA FIABILITÉ DE L'APPLICATION SHAREEVENT, UNE STRATÉGIE DE TESTS COMPLÈTE A ÉTÉ MISE EN PLACE. CETTE STRATÉGIE COMPREND À LA FOIS DES TESTS UNITAIRES ET DES TESTS D'INTÉGRATION, AVEC DES OUTILS TEL QUE POSTMAN .



6.2 TESTS UNITAIRES ET TESTS D'INTÉGRATION :

LES TESTS UNITAIRES ONT ÉTÉ UTILISÉS POUR VÉRIFIER LE BON FONCTIONNEMENT DES DIFFÉRENTES PARTIES DE L'APPLICATION DE MANIÈRE ISOLÉE. LES FONCTIONNALITÉS INDIVIDUELLES, TELLES QUE LA CRÉATION D'UN ÉVÉNEMENT, LA GESTION DES UTILISATEURS, L'AUTHENTIFICATION, ETC., ONT ÉTÉ TESTÉES EN UTILISANT DES FRAMEWORKS DE TEST TELS QUE JEST OU MOCHA.

LES TESTS UNITAIRES PERMETTENT DE S'ASSURER QUE CHAQUE COMPOSANT OU FONCTIONNALITÉ RÉPOND AUX EXIGENCES ET FONCTIONNE CORRECTEMENT, INDÉPENDAMMENT DES AUTRES PARTIES DE L'APPLICATION.

EN PLUS DES TESTS UNITAIRES, DES TESTS D'INTÉGRATION ONT ÉTÉ RÉALISÉS POUR VÉRIFIER LE BON FONCTIONNEMENT DE L'APPLICATION DANS SON ENSEMBLE. CES TESTS PERMETTENT DE VÉRIFIER L'INTERACTION ENTRE LES DIFFÉRENTES PARTIES DE L'APPLICATION ET DE S'ASSURER QUE LES DIFFÉRENTES FONCTIONNALITÉS FONCTIONNENT CORRECTEMENT ENSEMBLE.

DES BIBLIOTHÈQUES ET DES OUTILS APPROPRIÉS ONT ÉTÉ UTILISÉS POUR FACILITER LES TESTS, COMME LA BIBLIOTHÈQUE SUPertest POUR LES TESTS D'INTÉGRATION API, OU DES BIBLIOTHÈQUES DE TESTS POUR LES INTERFACES UTILISATEUR.

7.1 EVALUATION DES RESULTATS :

7.1 ÉVALUATION DES RÉSULTATS :

L'ÉVALUATION DES RÉSULTATS DE L'APPLICATION SHAREEVENT PEUT SE FAIRE À DIFFÉRENTS NIVEAUX. TOUT D'ABORD, IL EST IMPORTANT D'ÉVALUER LA CONFORMITÉ DE L'APPLICATION PAR RAPPORT AUX SPÉCIFICATIONS ET AUX FONCTIONNALITÉS DÉFINIES INITIALEMENT. LES TESTS EFFECTUÉS, QU'ILS SOIENT UNITAIRES, D'INTÉGRATION OU AVEC DES UTILISATEURS RÉELS, PERMETTENT DE S'ASSURER QUE L'APPLICATION RÉPOND AUX EXIGENCES ET FONCTIONNE DE MANIÈRE ATTENDUE.

ENSUITE, L'ÉVALUATION PEUT SE PORTER SUR LES PERFORMANCES DE L'APPLICATION. IL EST ESSENTIEL DE MESURER LES TEMPS DE RÉPONSE, LA SCALABILITÉ ET LA ROBUSTESSE DE L'APPLICATION POUR S'ASSURER QU'ELLE PEUT GÉRER UN NOMBRE CROISSANT D'UTILISATEURS ET DE CHARGES DE TRAVAIL.

ENFIN, L'ÉVALUATION DES RÉSULTATS PEUT ÉGALEMENT SE BASER SUR LES RETOURS DES UTILISATEURS RÉELS. LEURS COMMENTAIRES ET LEURS SUGGESTIONS PERMETTENT DE COMPRENDRE COMMENT L'APPLICATION EST PERÇUE ET UTILISÉE DANS UN CONTEXTE RÉEL. CES RETOURS SONT PRÉCIEUX POUR IDENTIFIER LES POINTS FORTS DE L'APPLICATION ET LES ASPECTS À AMÉLIORER.

7.2 LIMITATIONS ET DÉFIS RENCONTRÉS :

PENDANT LE DÉVELOPPEMENT DE L'APPLICATION SHAREEVENT, CERTAINES LIMITATIONS ET DÉFIS PEUVENT ÊTRE RENCONTRÉS. IL PEUT Y AVOIR DES CONTRAINTES DE TEMPS ET DE RESSOURCES QUI LIMITENT LA PORTÉE OU LA QUALITÉ DE L'APPLICATION. DES PROBLÈMES TECHNIQUES PEUVENT SURVENIR, TELS QUE DES BUGS DIFFICILES À RÉSOUDRE OU DES PROBLÈMES DE PERFORMANCE.

DE PLUS, LES ASPECTS DE SÉCURITÉ ET DE PROTECTION DES DONNÉES DOIVENT ÊTRE PRIS EN COMPTE. LES VULNÉRABILITÉS POTENTIELLES DOIVENT ÊTRE IDENTIFIÉES ET DES MESURES DE SÉCURITÉ APPROPRIÉES DOIVENT ÊTRE MISES EN PLACE POUR PROTÉGER LES UTILISATEURS ET LEURS INFORMATIONS.

NOTAMMENT LA RÉGLGLEMENTATION RGPD ,LA REGLEMENTATION DE LA CNIL .

D'UN POINT DE VUE LÉGAL IL EST IMPORANT DE NOUS RENSEIGNER SUR L'ASPECT STOCKAGE DE DONNÉE QU'ENGENDRE NOTRE APPLICATION .

7.3 PERSPECTIVES D'AMÉLIORATION :

ÉLARGIR LA PORTÉE DE L'APPLICATION : SHAREEVENT PEUT ÊTRE ÉTENDU POUR PRENDRE EN CHARGE D'AUTRES TYPES D'ÉVÉNEMENTS, TELS QUE DES ÉVÉNEMENTS PROFESSIONNELS, DES CONFÉRENCES OU DES FESTIVALS. CELA PERMETTRAIT D'ATTIRER UN PUBLIC PLUS LARGE ET DE DIVERSIFIER LES CAS D'UTILISATION DE L'APPLICATION.

INTÉGRATION DE FONCTIONNALITÉS DE GÉOLOCALISATION : LA POSSIBILITÉ D'INTÉGRER DES FONCTIONNALITÉS DE GÉOLOCALISATION PEUT ÊTRE ENVISAGÉE, CE QUI PERMETTRAIT AUX UTILISATEURS DE RECHERCHER DES ÉVÉNEMENTS DANS LEUR RÉGION OU DE VOIR LES ÉVÉNEMENTS À PROXIMITÉ.

MISE EN PLACE DE NOTIFICATIONS EN TEMPS RÉEL : L'AJOUT DE NOTIFICATIONS EN TEMPS RÉEL PEUT AMÉLIORER L'ENGAGEMENT DES UTILISATEURS EN LES TENANT INFORMÉS DES MISES À JOUR CONCERNANT LES ÉVÉNEMENTS AUXQUELS ILS SONT INTÉRESSÉS OU INSCRITS.

INTÉGRATION AVEC D'AUTRES PLATEFORMES ET RÉSEAUX SOCIAUX : IL PEUT ÊTRE BÉNÉFIQUE D'INTÉGRER SHAREEVENT AVEC D'AUTRES PLATEFORMES ET RÉSEAUX SOCIAUX POPULAIRES, TELS QUE FACEBOOK, INSTAGRAM OU LINKEDIN. CELA FACILITERAIT LE PARTAGE D'ÉVÉNEMENTS ET L'INTERACTION SOCIALE AUTOUR DES ÉVÉNEMENTS.

AMÉLIORATION DE L'ACCESSIBILITÉ : EN RENDANT L'APPLICATION ACCESSIBLE AUX PERSONNES HANDICAPÉES, NOTAMMENT EN METTANT EN PLACE DES FONCTIONNALITÉS D'ACCESSIBILITÉ POUR LES UTILISATEURS MALVOYANTS OU MALENTENDANTS, SHAREEVENT PEUT S'OUVRIR À UN PUBLIC PLUS LARGE ET GARANTIR UNE EXPÉRIENCE INCLUSIVE POUR TOUS LES UTILISATEURS.

AMÉLIORATION DES PERFORMANCES ET DE LA SCALABILITÉ : EN OPTIMISANT D'AVANTAGE LES PERFORMANCES DU SERVEUR, EN METTANT EN ŒUVRE UNE GESTION EFFICACE DES REQUÊTES ET EN OPTIMISANT L'UTILISATION DES RESSOURCES, L'APPLICATION PEUT GÉRER UNE CHARGE DE TRAVAIL CROISSANTE ET OFFRIR UNE EXPÉRIENCE FLUIDE MÊME DANS DES CONDITIONS DE TRAFIC ÉLEVÉ.

CES PERSPECTIVES D'AMÉLIORATION PERMETTRAIENT DE FAIRE ÉVOLUER L'APPLICATION SHAREEVENT, DE RÉPONDRE AUX BESOINS CHANGEANTS DES UTILISATEURS ET DE RESTER COMPÉTITIF SUR LE MARCHÉ DES APPLICATIONS D'ÉVÉNEMENTS. IL EST IMPORTANT DE PRENDRE EN COMPTE LES RETOURS DES UTILISATEURS ET DE SUIVRE LES ÉVOLUTIONS TECHNOLOGIQUES POUR CONTINUER À AMÉLIORER L'APPLICATION DANS LE TEMPS.

8 CONCLUSION

8.1 RÉSUMÉ DES CONTRIBUTIONS :

CE MÉMOIRE A PRÉSENTÉ SHAREEVENT, UNE APPLICATION PERMETTANT DE CRÉER DES ÉVÉNEMENTS ET DE MUTUALISER LES PHOTOS EN UN SEUL ENDROIT. LE DÉVELOPPEMENT DE L'APPLICATION A ÉTÉ RÉALISÉ EN UTILISANT REACT NATIVE POUR LA PARTIE FRONT-END MOBILE ET NESTJS AVEC TYPEORM POUR LA PARTIE BACK-END.

AU NIVEAU DE LA CONCEPTION, UNE MODÉLISATION DE LA BASE DE DONNÉES A ÉTÉ RÉALISÉE AVEC LUCIDCHART, SUIVANT LA MÉTHODE MERISE. DE PLUS, UNE MAQUETTE DE L'APPLICATION A ÉTÉ CONÇUE AVEC FIGMA, EN METTANT L'ACCENT SUR L'EXPÉRIENCE UTILISATEUR.

LE DÉVELOPPEMENT DE LA PARTIE FRONT-END A ÉTÉ DÉCRIT, EN METTANT EN AVANT L'UTILISATION DE REACT NATIVE POUR CRÉER LES FONCTIONNALITÉS DE L'APPLICATION.

LE DÉVELOPPEMENT DE LA PARTIE BACK-END A ÉGALEMENT ÉTÉ DÉTAILLÉ, EN SOULIGNANT L'UTILISATION DE NESTJS ET TYPEORM POUR METTRE EN PLACE LES FONCTIONNALITÉS DU SERVEUR.

DES TESTS ET VALIDATIONS ONT ÉTÉ EFFECTUÉS, COMPRENANT DES TESTS UNITAIRES, DES TESTS D'INTÉGRATION ET LA VALIDATION DE L'APPLICATION AVEC DES UTILISATEURS RÉELS.

ENFIN, DES PERSPECTIVES D'AMÉLIORATION ONT ÉTÉ PROPOSÉES, NOTAMMENT L'AJOUT DE NOUVELLES FONCTIONNALITÉS, L'AMÉLIORATION DES PERFORMANCES, LA SÉCURITÉ ET L'EXPÉRIENCE UTILISATEUR.

8.2 RETOUR SUR L'EXPÉRIENCE DE DÉVELOPPEMENT :

LE DÉVELOPPEMENT DE SHAREEVENT A ÉTÉ UNE EXPÉRIENCE ENRICHISSANTE. LES TECHNOLOGIES CHOISIES, REACT NATIVE ET NESTJS, ONT PERMIS DE DÉVELOPPER RAPIDEMENT UNE APPLICATION PERFORMANTE ET MULTIPLATEFORME. LA MÉTHODOLOGIE DE CONCEPTION UTILISÉE, BASÉE SUR LUCIDCHART ET MERISE, A FACILITÉ LA MODÉLISATION DE LA BASE DE DONNÉES ET LA COMPRÉHENSION DE LA STRUCTURE DE L'APPLICATION.

L'UTILISATION DE FIGMA POUR LA CONCEPTION DE LA MAQUETTE A PERMIS DE VISUALISER L'INTERFACE UTILISATEUR ET DE RECUEILLIR DES RETOURS PRÉCIEUX AVANT MÊME LE DÉVELOPPEMENT.

LES TESTS RÉALISÉS ONT PERMIS DE VÉRIFIER LE BON FONCTIONNEMENT DE L'APPLICATION À DIFFÉRENTS NIVEAUX ET DE GARANTIR SA QUALITÉ.

L'IMPLICATION DES UTILISATEURS RÉELS DANS LE PROCESSUS DE TEST ET DE VALIDATION A ÉTÉ PARTICULIÈREMENT BÉNÉFIQUE POUR AMÉLIORER L'APPLICATION ET RÉPONDRE AUX BESOINS RÉELS DES UTILISATEURS.

8.3 CONCLUSION GÉNÉRALE :

EN CONCLUSION, L'APPLICATION SHAREEVENT RÉPOND AU BESOIN DE CRÉER DES ÉVÉNEMENTS ET DE PARTAGER DES PHOTOS EN UN SEUL ENDROIT. GRÂCE À LA COMBINAISON DE TECHNOLOGIES MODERNES TELLES QUE REACT NATIVE ET NESTJS, L'APPLICATION OFFRE UNE EXPÉRIENCE UTILISATEUR FLUIDE ET UNE ARCHITECTURE ROBUSTE.

LES DIFFÉRENTES ÉTAPES DU DÉVELOPPEMENT, DE LA CONCEPTION À LA MISE EN ŒUVRE, ONT ÉTÉ DÉTAILLÉES, METTANT EN AVANT LES CHOIX TECHNOLOGIQUES ET LES MÉTHODOLOGIES UTILISÉES.

LES TESTS ET LES RETOURS DES UTILISATEURS ONT PERMIS DE VALIDER L'APPLICATION ET D'IDENTIFIER DES AXES D'AMÉLIORATION POUR DE FUTURES MISES À JOUR.

EN SOMME, SHAREEVENT OFFRE UNE SOLUTION CONVIVIALE ET EFFICACE POUR LA GESTION D'ÉVÉNEMENTS ET LE PARTAGE DE PHOTOS, ET OUVRE LA VOIE À DE NOUVELLES OPPORTUNITÉS D'INNOVATION DANS LE DOMAINE DES APPLICATIONS MOBILES.