

# GESTION DES ESPACES PUBLICS AU FESTIVAL ÉCHO



Ayyoub BOULAHRI

Yassine EL MESBAHY

Sofiya DEBOIS

Ilias REGUIG

Zakaria BOUMLINE

# Sommaire

01	PRÉSENTATION	page 3
02	MANUEL D'UTILISATION	page 4
03	STRUCTURE DU PROJET	page 5
04	TRAVAIL EFFECTUER	page 7
4-1	AYYOUB BOULAHRI	page 7
4-2	YASSINE EL MESBAHY	page 11
4-3	SOFIYA DEBOIS	page 15
4-4	ILIAS REGUIG	page 17
4-5	ZAKARIA BOUMLINE	page 21
05	MISE EN COMMUN DES TRAVAUX	page 11
06	SCÉNARIO D'UTILISATION	page 11
	CONCLUSION	page 11

# 01 Presentation

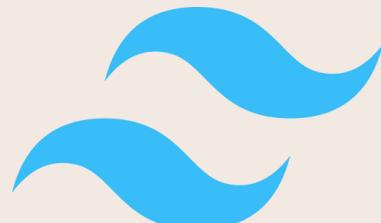
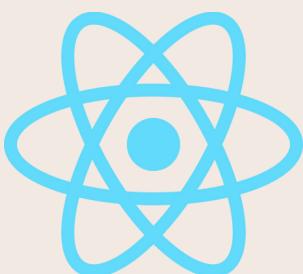
FestiSoif\_Website est une application web front-end conçue pour améliorer l'expérience des utilisateurs lors d'un événement de type festival. Le projet vise à proposer une interface moderne, intuitive et adaptée aux usages mobiles, permettant aux utilisateurs de s'orienter facilement, d'accéder à des informations utiles et d'interagir avec les différents services proposés sur le site.

Ce projet met l'accent sur la conception d'une architecture front-end claire et modulaire, facilitant la navigation entre les différentes pages et la réutilisation des composants. L'application est construite comme une Single Page Application (SPA), garantissant une navigation fluide sans recharge de page.

Afin de se concentrer sur le développement front-end, des fichiers de données statiques (mock data) sont utilisés pour simuler les échanges avec un backend. Cette approche permet de tester les flux utilisateurs, la navigation et l'affichage des données comme dans une application connectée à une API réelle, tout en gardant le projet simple et autonome.

## Choix Techniques :

- React pour la construction de composants réutilisables et la gestion de la navigation.
- JavaScript pour la logique applicative et la gestion des interactions utilisateur.
- Tailwind CSS pour le styling de l'interface, permettant de créer rapidement une UI cohérente.



02

# Manuel d'utilisateur

- Environnement de développement



- Node.js (version 16 ou supérieure) : pour exécuter le serveur de développement et construire l'application



- npm : pour gérer les dépendances du projet
- Git : pour cloner et gérer le dépôt du projet



```
● ● ●  
1 node -v  
2 npm -v  
3 git --version
```

- Navigateur moderne : Chrome (recommandé)



- Installation et lancement du projet

- Cloner le dépôt GitHub

```
● ● ●  
1 git clone https://github.com/Sofiya-D/FestiSoif_Website.git  
2 cd FestiSoif_Website
```

- Installer les dépendances

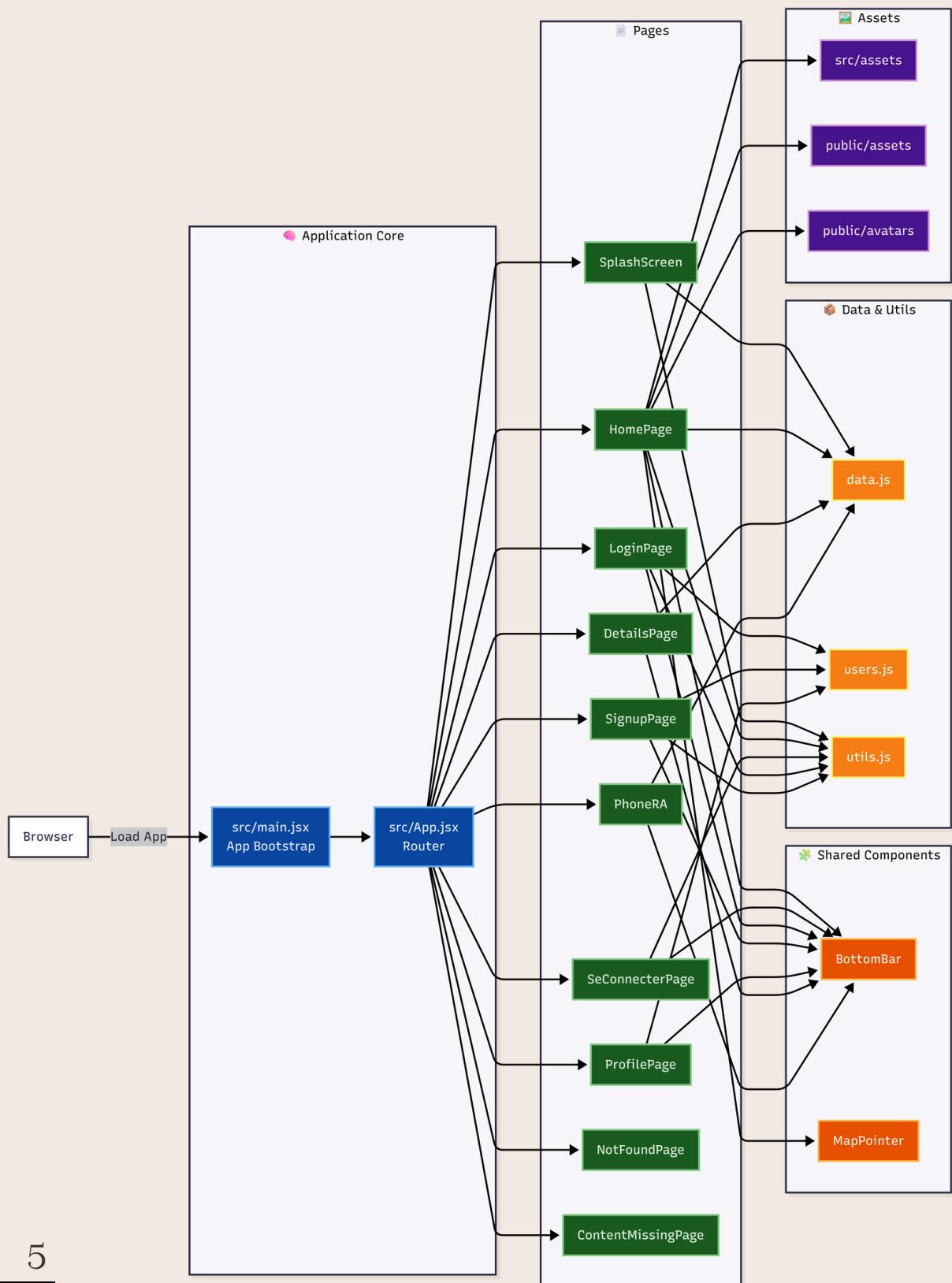
```
● ● ●  
1 npm install
```

- Lancer l'application en développement

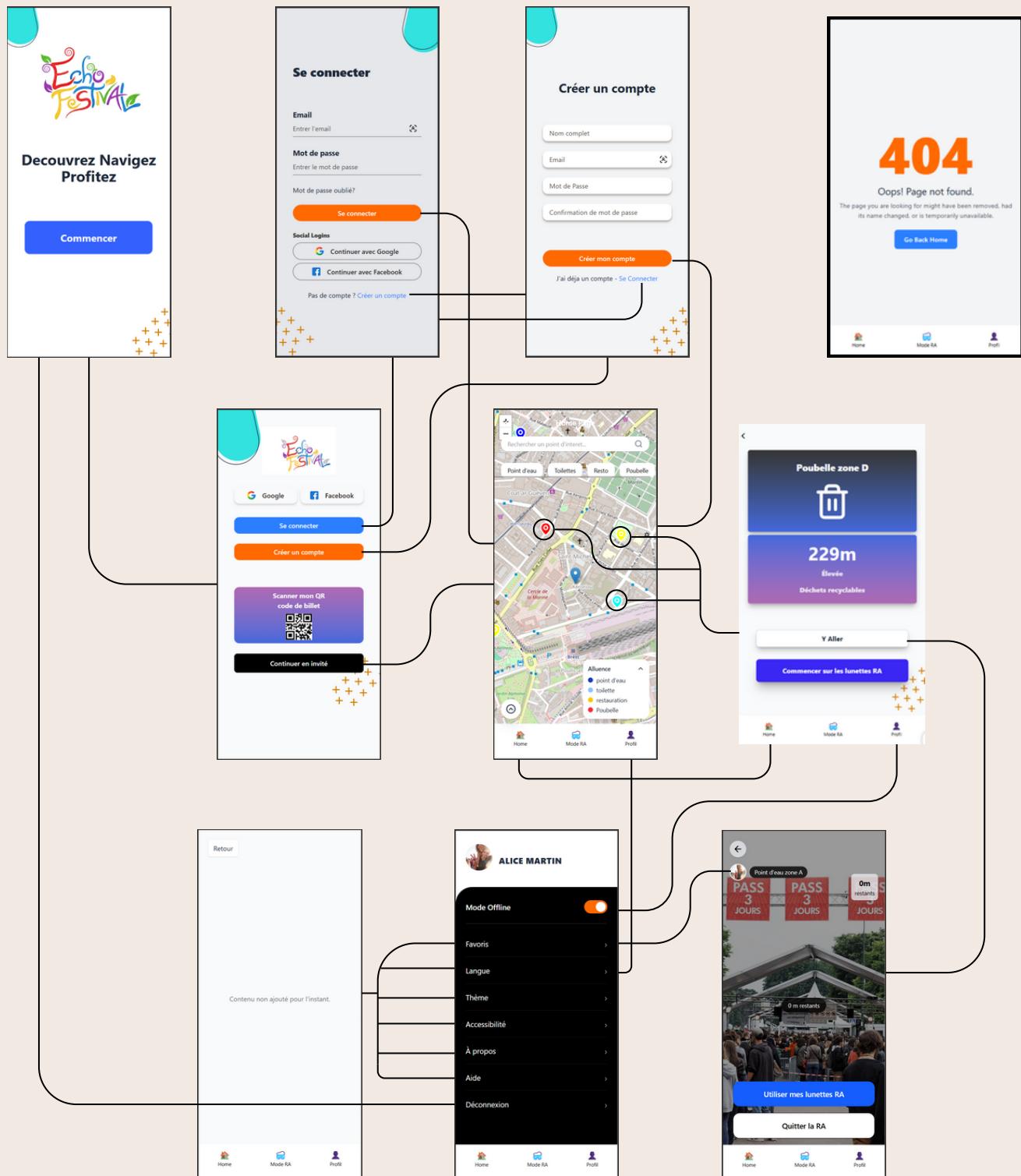
```
● ● ●  
1 npm run dev
```

03

# Structure du Projet



## • Navigation entre les pages



## 04 Travail Effectué

### 4-1 Ayyoub BOULAHRI

#### Introduction à mon travail

Dans le projet, j'ai participé activement au développement front-end et à la coordination de l'équipe. J'ai développé plusieurs pages clés de l'application :

- Page Map : carte interactive avec points d'intérêt filtrables et recherche dynamique
- Page PhoneRA : simulation d'une interface de réalité augmentée pour le festival
- Page NotFound : gestion des routes inexistantes pour améliorer l'expérience utilisateur

En tant que chef de projet, j'ai également :

- Aidé l'équipe à résoudre les problèmes techniques rencontrés
- Organisé et réparti les tâches quotidiennes
- Géré les conflits Git et assuré une collaboration fluide entre les membres

#### Page d'Accueil (/home)



#### Ce que j'ai voulu faire

L'objectif de cette page était de créer une page centrale et interactive permettant à l'utilisateur de visualiser le plan du festival, de localiser les points d'intérêt (points d'eau, toilettes, restauration, poubelles) et de rechercher facilement un service spécifique.

L'accent a été mis sur une expérience mobile fluide, avec une interface claire, des filtres simples et une interaction directe avec la carte.

## Ce que j'ai fait

D'un point de vue technique, j'ai développé une page Home basée sur React et Leaflet, intégrant plusieurs fonctionnalités clés :

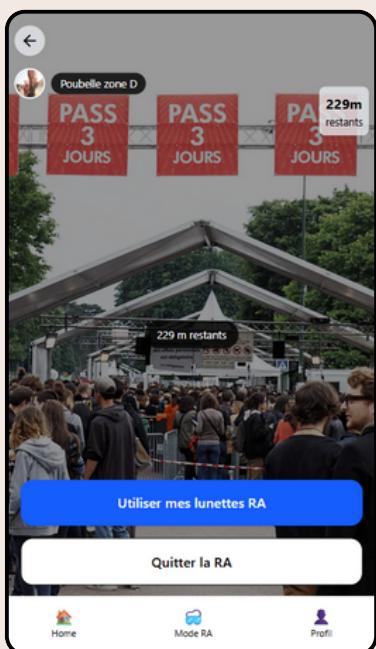
- Intégration d'une carte interactive à l'aide de react-leaflet, avec :
  - Un centre géographique prédéfini
  - Un niveau de zoom adapté à un usage mobile
  - La possibilité de changer le type de carte dynamiquement
- Gestion de l'état de l'application avec React hooks (useState) pour :
  - L'ouverture / fermeture de la légende
  - Le changement du fond de carte
  - La recherche textuelle
  - Les filtres par type de point d'intérêt
- Mise en place d'un système de recherche permettant de filtrer les points affichés sur la carte en fonction du nom saisi par l'utilisateur.
- Implémentation de filtres par catégorie (point d'eau, toilettes, restauration, poubelle), permettant d'afficher uniquement les points correspondant au type sélectionné.
- Utilisation de données statiques (points, maps, PointType) pour simuler une interaction avec un backend, tout en gardant une architecture facilement adaptable à une future API.
- Création et intégration du composant MapPointer, chargé d'afficher dynamiquement les marqueurs sur la carte à partir des données filtrées, et de rediriger l'utilisateur vers la page /detail en cliquant sur un marqueur, avec les informations correspondant au point sélectionné.
- Ajout d'éléments d'interface utilisateur :
  - Barre supérieure avec navigation
  - Barre de recherche
  - Boutons de filtres horizontaux
  - Légende interactive
  - Bouton de changement de vue de la carte

## Ce qui pourrait être fait

Plusieurs améliorations pourraient être envisagées :

- Connexion à un backend réel pour récupérer les points d'intérêt dynamiquement
- Ajout d'un système de géolocalisation pour afficher la position de l'utilisateur
- Ajout d'un système de clusters pour gérer un grand nombre de marqueurs

## Page RA (/phonera)



## Ce que j'ai voulu faire

L'objectif de cette page était de simuler une expérience en réalité augmentée permettant à l'utilisateur de visualiser un point d'intérêt en utilisant la caméra de son appareil, avec des informations contextuelles (distance, nom du point) et des options d'interaction.

L'idée était de créer une interface immersive, avec un affichage clair et lisible même sur mobile.

## Ce que j'ai fait

j'ai développé la page en React, avec plusieurs fonctionnalités clés :

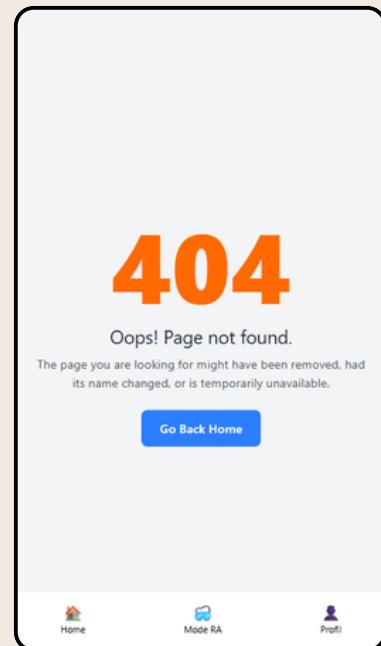
- Accès à la caméra de l'utilisateur via `navigator.mediaDevices.getUserMedia` pour afficher le flux vidéo en arrière-plan.
- Affichage d'un background statique et d'un overlay transparent pour améliorer la lisibilité des informations.
- Intégration des informations du point d'intérêt (nom, distance) provenant des données statiques (`points.js`), pour simuler un backend.
- Ajout de boutons interactifs :
  - “Utiliser mes lunettes RA”
  - “Quitter la RA”
  - Bouton retour pour naviguer vers la page précédente
- Affichage du nom du point d'intérêt et de l'avatar de manière visible et responsive.
- Positionnement dynamique de la distance restante au centre de l'écran et dans un badge en haut à droite.

## Ce qui pourrait être fait

- Intégrer une réalité augmentée réelle en utilisant une librairie AR (ex : ARjs ou Three.js) pour superposer des éléments 3D sur le flux vidéo.
- Ajouter une navigation interactive pour guider l'utilisateur vers le point d'intérêt.
- Améliorer le design et les animations pour une expérience plus immersive.
- Connexion à un backend pour récupérer dynamiquement les points et distances en temps réel.

## NotFound Page

Cette page gère les routes inexistantes afin d'améliorer l'expérience utilisateur. Elle affiche un message clair indiquant que la page recherchée n'existe pas, avec un bouton permettant de revenir facilement à la page d'accueil. La page est simple, responsive et stylée avec Tailwind CSS, garantissant une navigation fluide même en cas d'erreur de route.



## Difficultés rencontrées

- Au niveau global, les principales difficultés ont été la coordination entre les pages et composants, l'adaptation de l'interface pour un usage mobile, la simulation des données pour remplacer un backend réel, ainsi que la gestion des interactions complexes sur la carte et le flux vidéo tout en conservant une interface claire et performante.
- Nous avons également rencontré des conflits Git lors de l'intégration des modifications de plusieurs membres, nécessitant des résolutions manuelles pour maintenir la cohérence du projet.

## 4-2 Yassine El Mesbahy

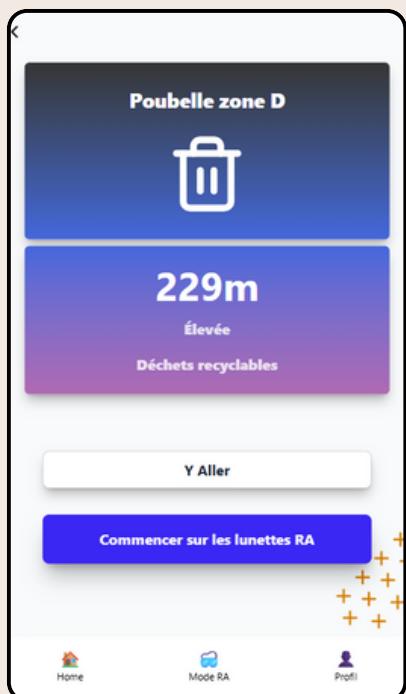
### Introduction à mon travail

Dans la continuité du travail présenté précédemment sur la page d'accueil et la carte interactive, ma contribution commence après la sélection d'un point d'intérêt par l'utilisateur.

Une fois qu'un point est sélectionné sur la carte, un identifiant est transmis via l'URL vers la page de détails. Mon travail consiste alors à récupérer cet identifiant, à traiter les données associées et à afficher les informations détaillées de la ressource sélectionnée.

J'ai également réalisé la page de lancement (Splash Screen) affichée au démarrage de l'application.

### Page de détails d'une ressource



### Présentation de la page

La page de détails est accessible via une route dynamique de la forme : « /details/:id »

Cette page est affichée après la sélection d'un point d'intérêt sur la carte. L'identifiant de la ressource est transmis dans l'URL par la page précédente, puis récupéré afin d'afficher les informations correspondantes.

### Objectif de la page

L'objectif principal de cette page est de fournir à l'utilisateur une vue détaillée de la ressource sélectionnée, tout en lui proposant des actions pour s'y rendre, notamment via la navigation en réalité augmentée.

## Fonctionnement et données utilisées

Les informations affichées sur cette page proviennent d'un tableau de données JavaScript local.

Chaque ressource est représentée par un objet contenant plusieurs propriétés, telles que :

- un identifiant (id) ;
- un nom (name) ;
- un type de ressource (type) ;
- une description (desc) ;
- une position géographique (position) ;
- un niveau d'occupation (occupation).

Lors du chargement de la page, l'identifiant est récupéré depuis l'URL, puis utilisé pour retrouver la ressource correspondante dans le tableau de données. Les différentes propriétés de cet objet sont ensuite affichées dynamiquement dans l'interface.

## Informations affichées

La page de détails affiche les éléments suivants :

- le nom de la ressource ;
- une icône représentant le type de ressource (point d'eau, toilettes, restauration ou poubelle) ;
- la distance entre l'utilisateur et la ressource ;
- le niveau d'occupation ;
- une description courte de la ressource.

L'icône est sélectionnée automatiquement en fonction du type de ressource afin de faciliter la compréhension visuelle pour l'utilisateur.

## Boutons et navigation



Deux boutons sont proposés sur cette page :

- "Y aller" : ce bouton permet de naviguer vers une page dédiée à la navigation en réalité augmentée (PhoneRA). L'identifiant de la ressource et la distance sont transmis lors de cette navigation.
- "Commencer sur les lunettes RA" : ce bouton est actuellement présent dans l'interface, mais la fonctionnalité permettant de lancer directement la navigation en réalité augmentée n'est pas encore implémentée.

## Difficultés rencontrées

Les principales difficultés rencontrées lors de ce travail sont :

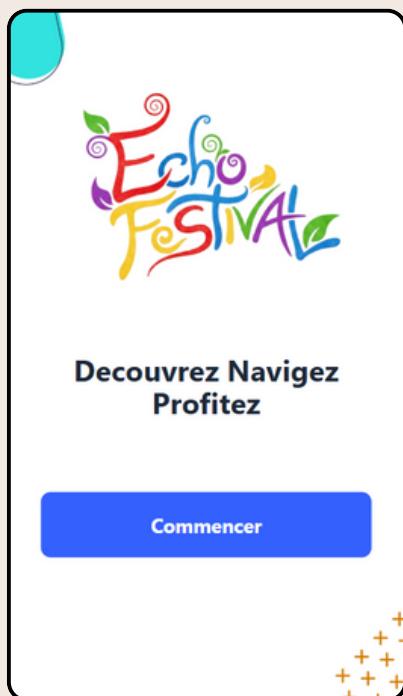
- la gestion des routes dynamiques et des paramètres dans l'URL ;
- la récupération correcte de l'identifiant transmis depuis la page précédente ;
- l'organisation des données afin d'afficher les informations correspondant à la bonne ressource.

## Améliorations possibles

Plusieurs améliorations peuvent être envisagées :

- remplacer les données locales par une API distante ;
- rendre fonctionnel le bouton "Commencer sur les lunettes RA" ;
- améliorer l'ergonomie et l'interface utilisateur ;
- ajouter des animations pour rendre la navigation plus fluide.

## Splash Screen



### Présentation de la page

La Splash Screen est la première page affichée lors du lancement de l'application, avant l'accès aux fonctionnalités principales.

### Objectif de la page

L'objectif de cette page est de proposer une entrée simple et claire dans l'application, en guidant l'utilisateur vers la page d'authentification sans surcharge d'informations.

## Fonctionnement

La page affiche :

- le logo de l'application ;
- un slogan ;
- un bouton "Commencer".

Lorsque l'utilisateur clique sur ce bouton, il est redirigé vers la page de connexion grâce au système de navigation de l'application.

## Améliorations possibles

Des améliorations peuvent être envisagées, comme :

- ajouter une animation ou une redirection automatique après quelques secondes ;
- améliorer le design visuel ;
- adapter le contenu selon le type d'utilisateur.

## Conclusion personnelle

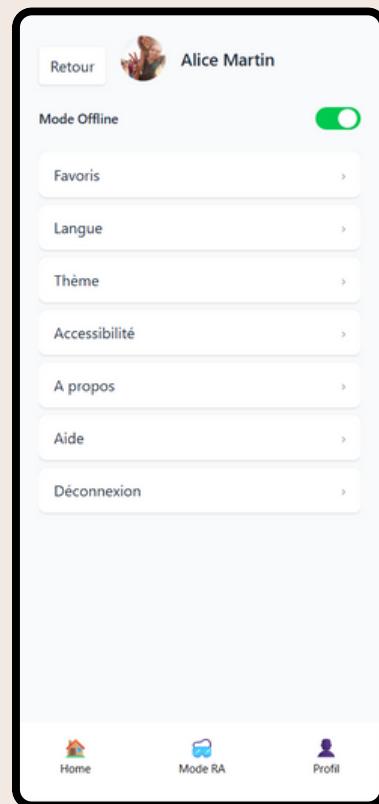
Cette partie du projet m'a permis de travailler sur la continuité du parcours utilisateur après la sélection d'un point sur la carte. J'ai pu approfondir l'utilisation du routage dynamique, la gestion des paramètres d'URL et l'affichage dynamique des données. Ce travail constitue une base solide pour l'évolution future de l'application, notamment pour l'intégration complète de la navigation en réalité augmentée.

## 4-3 Sofiya DEBOIS

### Ce que j'ai voulu faire

Je me suis chargée de coder la page Profil Utilisateur.

Cette page devait essentiellement être un index pour accéder à différentes sous-pages d'information et de réglages; et permettre l'activation du mode offline. J'ai donc réalisé une page assez simpliste reprenant les fonctionnalités attendues



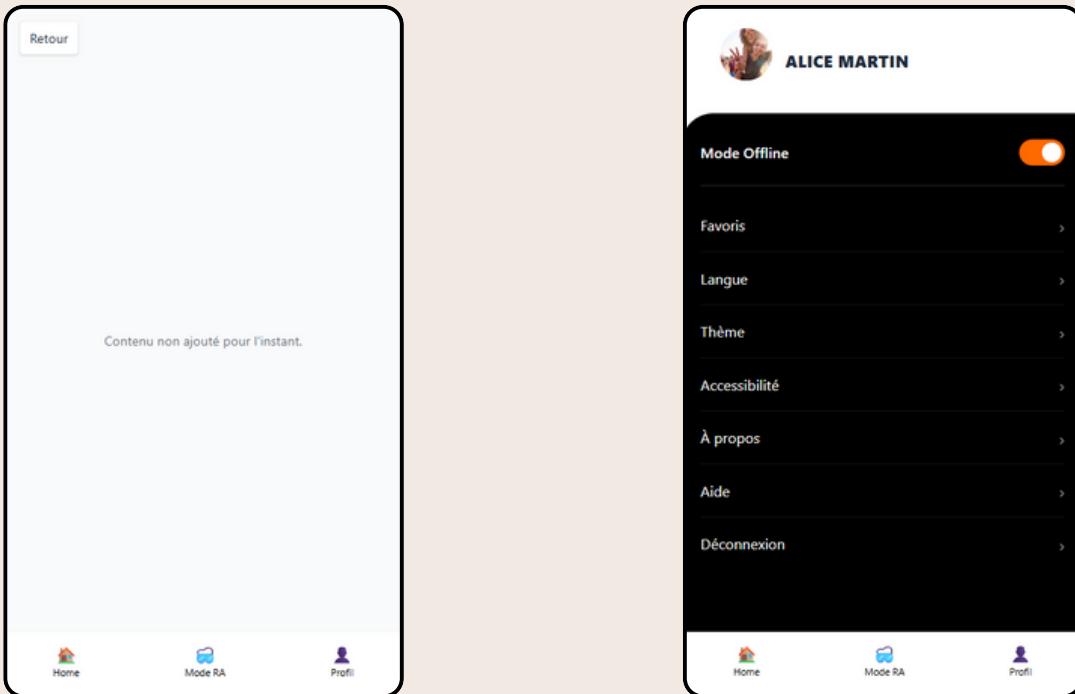
### Ce que j'ai fait

La photo de profil et le nom d'utilisateur s'adaptent aux données de l'utilisateur connecté. J'ai rajouté une page «oups» (Capture 2 ci-dessous) à laquelle on accède lorsqu'on clique sur les différentes sections, afin de préserver l'interactivité pensée pour la page profil sans m'attarder sur les fonctionnalités additionnelles que ces sections prévoient. Cette page «oups» contient juste une petite phrase et un bouton retour pour permettre une navigation fluide.

Je n'ai initialement pas prêté grande attention à l'apparence des pages; afin de privilégier l'ajout des fonctionnalités attendues.

Ayyoub m'a aidée à ajouter les fonctionnalités de photo de profil et nom d'utilisateur adaptatifs, car je ne savais pas comment faire et il était plus rapide de faire appel à l'aide de mes camarades que de chercher sur internet. Ayyoub a aussi retravaillé la page profil entre deux séances, afin que son apparence corresponde mieux à celle du prototype Figma (Capture 3 ci-dessous).

En dehors du code, je me suis chargée de rédiger le cahier des charges dans le Readme (en me basant sur notre Figma); et créer et attribuer quelques tâches sommaires aux membres du groupe dans le projet GitHub. J'ai donc passé un peu moins de temps à coder.



## Ce qui pourrait être fait

Cette partie du projet pourrait être améliorée en complétant les pages et fonctionnalités prévues par les sections de la page profil.

## 4-4 Ilias REGUIG

### Introduction à mon travail

Dans la continuité du travail sur la page d'accueil, la carte interactive et les pages de détails des ressources, ma contribution se concentre sur l'authentification des utilisateurs et la personnalisation de l'expérience utilisateur.

Mon travail comprend :

- La création des pages d'authentification (connexion et inscription)
- La refonte de l'interface de connexion existante
- L'ajout d'éléments visuels décoratifs et de personnalisation (étoiles et blobs)
- L'implémentation de la logique d'authentification
- La gestion des états de session utilisateur

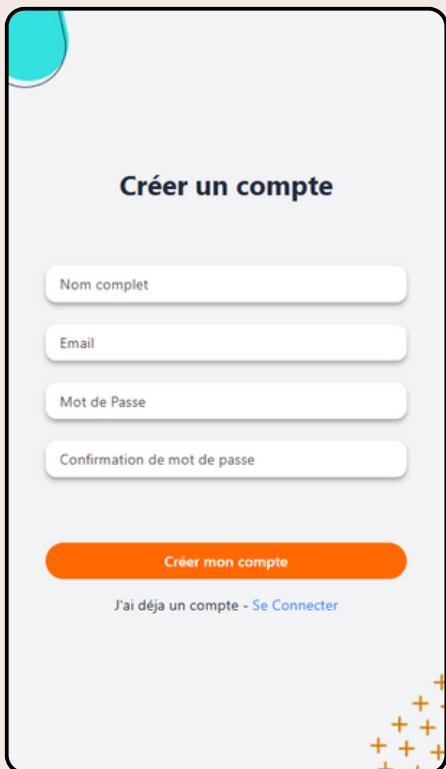
### Page d'inscription (/signup)

#### Présentation de la page

La page d'inscription permet aux nouveaux utilisateurs de créer un compte pour accéder aux fonctionnalités personnalisées de l'application.

L'interface comprend :

- Champ « Nom complet » : pour saisir le nom de l'utilisateur
- Champ « Email » : pour l'adresse électronique
- Champ « Mot de passe » : pour définir un mot de passe sécurisé
- Champ « Confirmation de mot de passe » : pour vérifier la saisie du mot de passe
- Un bouton "Créer mon compte" pour valider l'inscription
- Un lien "J'ai déjà un compte - Se Connecter" pour rediriger vers la page de connexion



## Page de connexion (/seconnecter)



### Présentation de la page

La page de connexion permet aux utilisateurs existants d'accéder à leur compte, avec plusieurs options d'authentification.

L'interface comprend plusieurs sections :

- Champ Email avec le placeholder "Entrer l'email"
- Champ Mot de passe avec le placeholder "Entrer le mot de passe"
- Lien "Mot de passe oublié?" pour la récupération de compte
- Bouton "Se connecter" pour valider l'authentification

Section d'authentification sociale (pas intégré avec les API Google et Facebook) :

- Option "Continuer avec Google"
- Option "Continuer avec Facebook"

Section d'inscription :

- Lien "Pas de compte ? Créez un compte" pour rediriger vers la page d'inscription

## Refonte de l'interface existante

J'ai remodelé la page de connexion existante (/login) pour :

- Harmoniser le design avec les autres pages de l'application
- Améliorer l'ergonomie et la lisibilité des éléments
- Optimiser l'expérience mobile avec un design responsive
- Décorations artistiques (étoiles et motifs)

## Personnalisation de l'expérience utilisateur

Images de profil :

J'ai implémenté un système d'images de profil personnalisées :

- Différentes images de profil pour différents types d'utilisateurs présent dans un fichier JSON pour avoir une variété d'utilisateurs.
- Attribution automatique d'une image de profil dans le cas du connexion Guest
- Possibilité future d'upload d'image personnelle

Décorations et style artistique

- Ajout d'éléments décoratifs (étoiles) sur toutes les pages du site web
- Style artistique cohérent avec l'identité visuelle de l'application

## Logique d'authentification back-end

Création de compte :

- J'ai développé la logique permettant la connexion tant qu'un utilisateur déjà existant dans le fichier JSON « users.js » à partir de /seconnecter, aussi que la possibilité de soit créer son propre compte et accéder à l'application depuis /signup ou la possibilité de faire le login comme Guest qui créer automatiquement un user null suivant le format JSON.

Gestion de session:

- Stockage des données utilisateur dans le localStorage (alternative simple aux cookies complexes)
- Persistance de la session entre les rafraîchissements de page

## Gestion de la navigation et des routes

Restriction d'affichage de la navbar

J'ai implémenté une logique pour empêcher l'affichage de la barre de navigation sur certaines routes :

- /seconnecter : page de connexion
- /signup : page d'inscription
- Autres pages d'authentification si nécessaire

Cette approche permet :

- Une expérience plus épurée sur les pages d'authentification
- Éviter la distraction des utilisateurs pendant le processus d'authentification
- Renforcer la sécurité en limitant l'accès aux fonctionnalités sans authentification

### Difficultés rencontrées

- J'ai rencontré quelques défis lors de l'intégration des pages d'authentification, notamment dans la résolution de certains conflits avec le code existant et dans le maintien d'une cohérence visuelle avec le design original de l'application pour autres pages.
- Assurer une transition fluide entre les différentes pages tout en conservant l'identité visuelle globale a nécessité des ajustements minutieux.

### Améliorations possibles

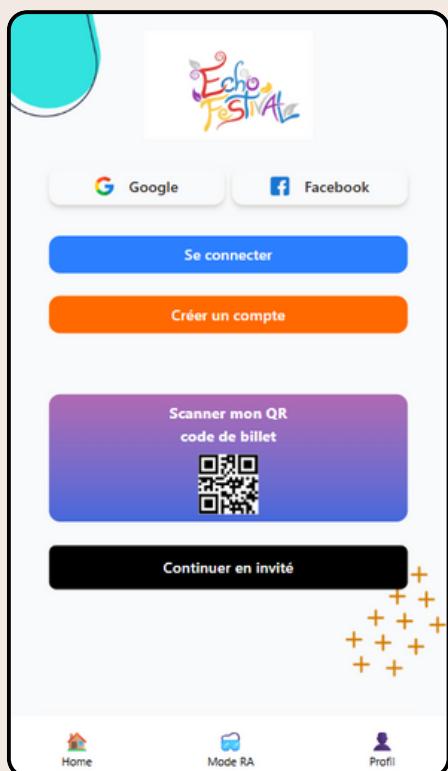
- Plusieurs améliorations peuvent être envisagées, telles que l'intégration avec une API back-end réelle pour une authentification plus sécurisée des utilisateurs, et aussi l'intégration avec des API pour pouvoir faire la connexion avec Google et Facebook, l'implémentation d'un système de récupération de mot de passe, l'ajout d'une authentification à deux facteurs, et l'amélioration des animations de transition pour une expérience utilisateur plus fluide.

## 4-5 Zakaria BOUMLINE

### Introduction à mon travail

Je me suis chargée de concevoir et de développer la page de connexion (Login), qui permet à l'utilisateur d'accéder à l'application soit via un compte personnel, soit en tant qu'invité. Cette page constitue un point d'entrée central dans le parcours utilisateur, notamment après la page de lancement.

### Page de login



### Interface utilisateur

L'interface de la page de connexion comprend :

- le logo de l'application ;
- des boutons de connexion via des services externes (Google et Facebook) ;
- un bouton Se connecter ;
- un bouton Créer un compte ;
- une option permettant de continuer en tant qu'invité ;
- une fonctionnalité de scan de QR code pour accéder à l'application via un billet.

L'interface a été conçue en respectant la charte graphique définie dans le prototype Figma, afin de garantir une expérience utilisateur cohérente et intuitive.

### Fonctionnement de la page

Le fonctionnement de la page de connexion repose sur une navigation simple entre plusieurs écrans de l'application.

Lorsque l'utilisateur interagit avec l'un des boutons, il est redirigé vers la page correspondante :

- le bouton Se connecter redirige vers la page d'authentification ;
- le bouton Créer un compte permet d'accéder à la page d'inscription ;
- l'option Continuer en tant qu'invité donne accès à l'application sans authentification ;
- le bouton de scan de QR code permet de déclencher une fonctionnalité d'accès rapide à l'application.

## Difficultés rencontrées

Le développement de cette page a constitué ma première expérience complète de création d'une interface web, ce qui a rendu cette étape plus complexe que prévu.

Les principales difficultés rencontrées concernent :

- la mise en place de la navigation entre les pages, notamment la gestion du basculement d'une page à une autre via les fonctionnalités de navigation ::
- la prise en main du fonctionnement de React, notamment la logique des composants et leur organisation.

## Améliorations possibles

Plusieurs améliorations peuvent être envisagées pour cette page :

- ajouter des animations pour rendre l'interface plus dynamique ;
- améliorer l'ergonomie générale de la page.

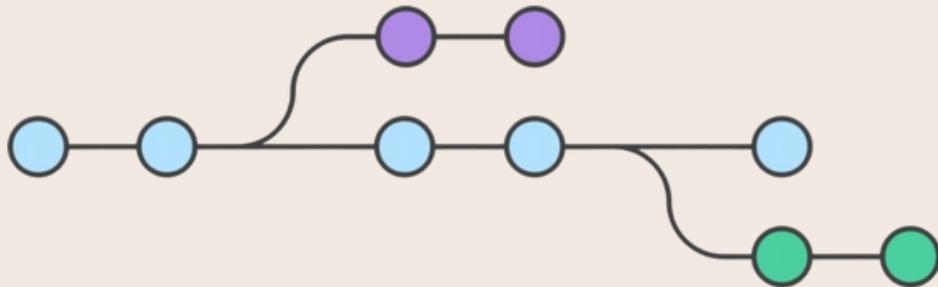
05

## Mise en commun des travaux

Le développement du projet s'est organisé autour d'un travail en branches séparées, permettant à chacun de travailler de manière autonome sur ses fonctionnalités assignées. Chaque membre a ainsi développé dans son propre environnement avant de procéder à l'intégration finale sur la branche principale.

Lors de la phase de fusion, quelques conflits sont naturellement apparus, principalement liés à des chevauchements dans les routes, les composants ou les styles. Ces conflits ont nécessité une attention particulière pour maintenir la cohérence du projet et coordonner les modifications entre les membres du groupe.

Au final, le processus s'est déroulé de manière fluide et sans obstacles majeurs, grâce à une bonne organisation en amont et à une communication efficace au sein de l'équipe. L'application a pu être intégrée avec succès, combinant harmonieusement les contributions de chacun.



06

# Scénario d'utilisation

## Premier lancement de l'application

L'utilisateur accède à l'application pour la première fois via son téléphone.

Il trouve des Écrans successifs :

- Splash Screen : Affichage du logo et du slogan de l'application pendant quelques instants.
- Redirection automatique vers la page de connexion principale.

## Choix d'authentification

Sur la page de connexion, l'utilisateur a plusieurs options :

- Option A : Créer un compte
  - Clique sur "Créer un compte"
  - Redirection vers /signup
  - Remplit le formulaire (nom, email, mot de passe)
  - Compte créé et stocké dans localStorage
  - Redirection vers la page d'accueil
- Option B : Se connecter avec un compte existant
  - Clique sur "Se connecter"
  - Redirection vers /seconnecter
  - Saisit ses identifiants (email/mot de passe)
  - Authentification réussie depuis le fichier JSON
  - Redirection vers la page d'accueil
- Option C : Continuer en tant qu'invité
  - Clique sur "Continuer en tant qu'invité"
  - Crédit automatique d'un profil temporaire
  - Redirection immédiate vers la page d'accueil

## Navigation sur la page d'accueil

L'utilisateur se retrouve sur la page d'accueil qui présente :

- La carte interactive avec les différents points d'intérêt.
- La barre de navigation permettant d'accéder aux différentes sections.
- La possibilité de visualiser sa position sur la carte.

## Sélection d'un point d'intérêt

L'utilisateur :

- Navigue sur la carte interactive.
- Repère un point d'intérêt (toilettes, point d'eau, restauration, poubelle).
- Clique sur le marqueur correspondant.
- L'application transmet l'identifiant du point via l'URL.

## Consultation des détails de la ressource

Redirection automatique vers la page de détails details/:id :

- Affichage des informations complètes de la ressource (nom, type, description).
- Visualisation du niveau d'occupation en temps réel.
- Calcul et affichage de la distance depuis la position de l'utilisateur.
- Affichage de l'icône correspondant au type de ressource.

## Choix du mode de navigation

Depuis la page de détails, l'utilisateur peut :

- Option A : Navigation standard
  - Clique sur "Y aller"
  - Redirection vers la page de navigation en réalité augmentée (PhoneRA)
  - Affichage des instructions de cheminement
- Option B : Navigation RA (à implémenter)

## Accès au profil utilisateur

L'utilisateur peut à tout moment :

- Cliquer sur l'icône de profil dans la barre de navigation.
- Accéder à la page profil.
- Visualiser ses informations personnelles (nom, photo de profil).
- Accéder aux sections de réglages (page "oups" pour l'instant).
- Activer/désactiver le mode hors ligne.

## Déconnexion

L'utilisateur décide de quitter l'application :

- Accède à la page profil.
- Sélectionne l'option de déconnexion.
- Session supprimée du localStorage.
- Redirection vers la page de connexion principale.

## Cas d'usage typique

un étudiant participant à un festival de musique pour le week-end, ouvre l'application sur son téléphone en arrivant sur le site bondé. Il se connecte en tant qu'invité pour éviter les formalités et accède instantanément à la carte interactive du festival. Cherchant rapidement des toilettes pour se rafraîchir, il repère l'icône correspondante la plus proche sur la carte, la sélectionne et consulte la page de détails qui lui indique une distance de 150 mètres et un faible taux d'occupation. Il utilise la fonction "Y aller" pour être guidé à travers la foule. Plus tard, assoiffé, il recherche un point d'eau, sélectionne à nouveau le plus proche sur la carte, et suit les instructions de navigation pour s'y rendre, utilisant l'application tout au long de la journée pour localiser les stands de nourriture, les poubelles de tri et différentes scènes, optimisant ainsi son expérience festive sans perdre de temps.

# Conclusion



Le projet a permis de mettre en pratique nos compétences en développement web, en particulier avec React et Tailwind CSS, et d'expérimenter la gestion d'un projet en équipe. Nous avons réussi à créer des pages interactives, tout en simulant un backend avec des fichiers statiques.

Cependant, certaines limites sont apparues :

- L'absence d'un backend réel limite l'interactivité et la mise à jour dynamique des données.
- La réalité augmentée reste une simulation et n'exploite pas pleinement les capacités des appareils mobiles.
- Des conflits Git et des problèmes de coordination ont parfois ralenti l'avancement du projet.

Malgré ces limites, le projet constitue une base solide et fonctionnelle, permettant de visualiser clairement les fonctionnalités principales et de préparer de futures améliorations techniques et UX. Il souligne également l'importance de la planification, de la collaboration et de la gestion des conflits dans un projet de groupe.

