Documentation technique PARKFINDER

**Partie Front-end**

Pour gérer la partie Frontend on a utilisé comme technologie React

***React c’est quoi ?***

React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état

Pour générer un projet react on doit lancer cette commande on lui donne le nom de notre projet dans notre ça c’est ***client***

**$ npx create-react-app client**

Après avoir créer le projet on entre dans le projet avec la ligne de commande avec cette commande

**$ cd client**

Ensuite on lance la commande npm start to run le projet par defaut le port utilisé dans React c’est le port 3000

**$ npm start**

Voici l’architecture du projet React

Une image contenant texte

Description générée automatiquement

Nous avons créer un dossier *Components* qui contient plusieurs dossiers chaque dossier contient des fichier *JSX* qui correspond à une page dans notre application

Le fichier app.js contient la config du *routage* de notre application

Fichier *firebase.js* contient les imports nécessaires qu’on a utilisé pour appeler les fonctions de Firebase

Fichier *index.js* c’est ou on affiche les composants de notre app on se basant bien sûr sur l’url

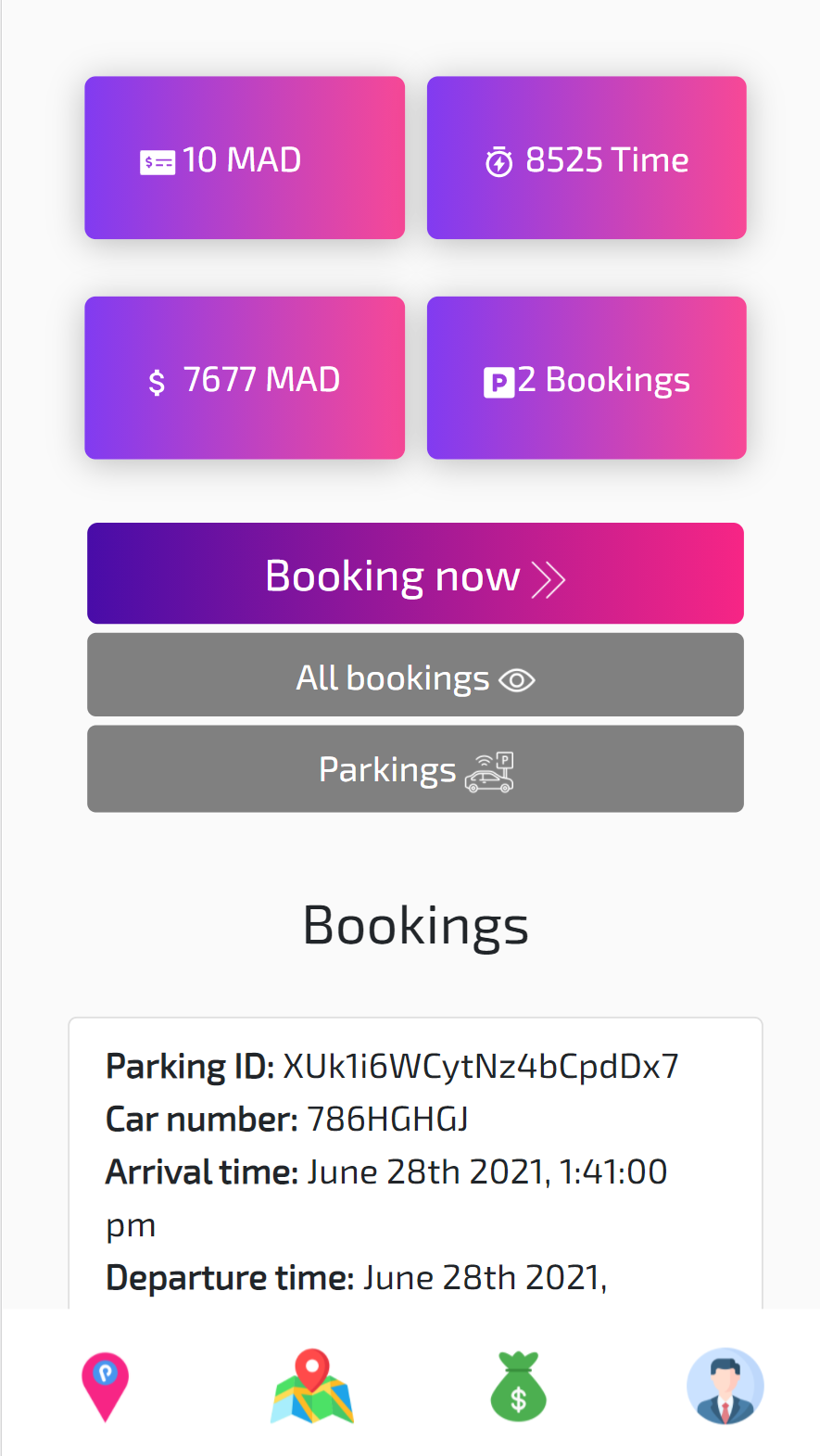
Dossier public contient un dossier images qui contient les images qu’on a utiliser dans notre application ainsi le nôtre page index.html et le *manifest.json* et le *worker.js* pour travailler avec le **PWA**

**Voici les dépendances de notre application / modules installer**

Une image contenant texte

Description générée automatiquement

Voici le résultat sur le navigateur :



**Partie Authentification avec Firebase**

Pour commencer à utiliser Firebase as a backend nous devons installer Firebase dans notre projet React

Installez le SDK JavaScript Firebase :

1. Si vous n'avez pas encore de package.jsonfichier, créez-en un en exécutant la commande suivante à partir de la racine de votre projet JavaScript :

**$ npm init**

Installez le firebase package npm et enregistrez-le dans votre package.json fichier en exécutant :

**$ npm install –save firebase**

Ensuite on créer un fichier firebase.js qui va contenir le SDK de projet firebase pour qu’il connaitre notre projet React ainsi les imports dans on va utiliser pour utiliser les fonctions de firebase

Une image contenant texte

Description générée automatiquement

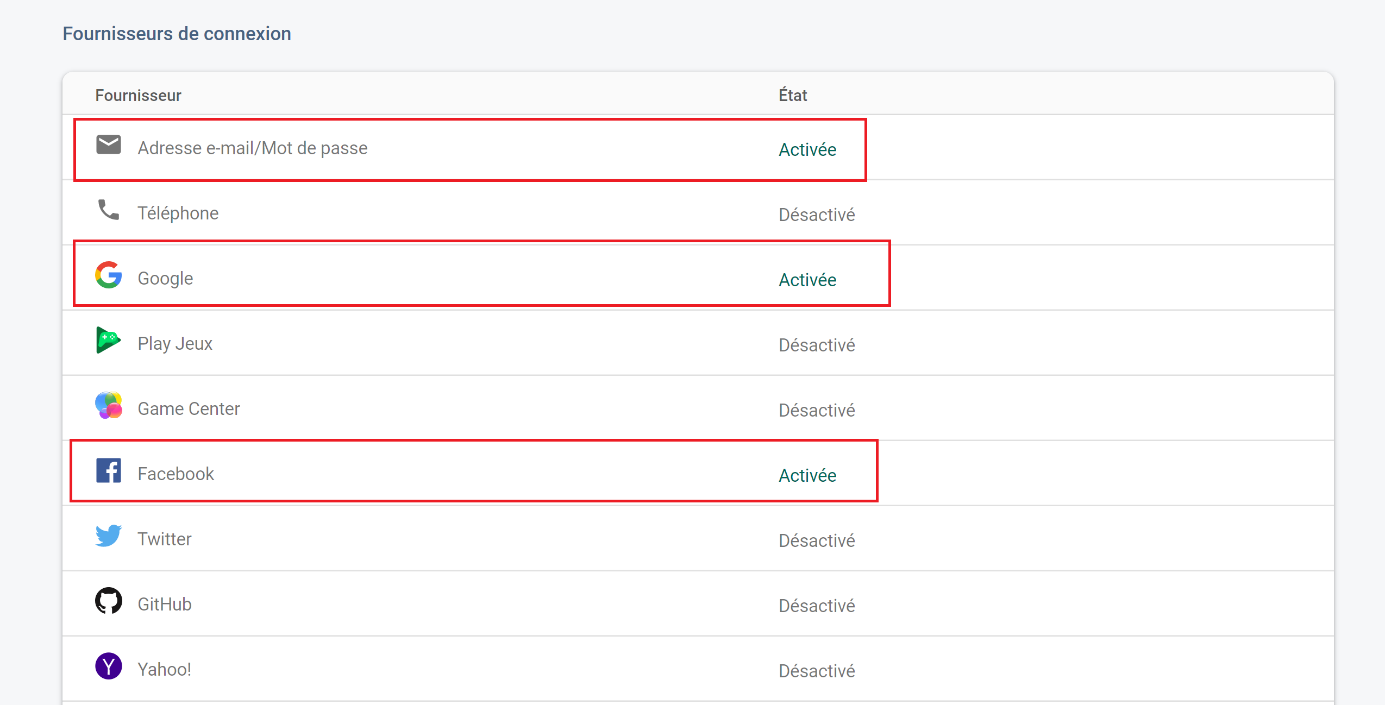
Pour avoir la SDK du firebase on se rendre dans la console du firebase après on accéde a notre projet ensuite on va sur paramétre du projet et on copier la SDK

Une image contenant texte

Description générée automatiquement

Nous avons utilisé l'authentification Firebase pour permettre aux utilisateurs de se connecter à notre application à l'aide des plusieurs méthodes de connexion, y compris la connexion par adresse e-mail et mot de passe, et la connexion avec Google Sign-in et Facebook Login ce qu’on appelle les fournisseurs d'identité fédérés

Avant de procéder a travailler avec l’authentification de Firebase nous devons activer sur la console de Firebase authentication avec Email/Password / Facebook / Google



Pour créer un nouveau compte utilisateur avec un mot de passe, procédez comme suit sur la page d'inscription de votre application :

1. Lorsqu'un nouvel utilisateur s'inscrit à l'aide du formulaire d'inscription de votre application, effectuez toutes les étapes de validation de nouveau compte requises par votre application, par exemple en vérifiant que le mot de passe du nouveau compte a été correctement saisi et répond à vos exigences de complexité.
2. Créez un nouveau compte en transmettant l'adresse e-mail et le mot de passe du nouvel utilisateur à createUserWithEmailAndPassword :

Une image contenant texte

Description générée automatiquement

**Connecter un utilisateur avec une adresse e-mail et un mot de passe**

Les étapes de connexion d'un utilisateur avec un mot de passe sont similaires aux étapes de création d'un nouveau compte. Sur la page de connexion de votre application, procédez comme suit :

1. Lorsqu'un utilisateur se connecte à votre application, transmettez son adresse e-mail et son mot de passe à signInWithEmailAndPassword :

Une image contenant texte

Description générée automatiquement

On redirige l’utilisateur dans la page FileManager s’il a entré les bons email and mot de passe

**Connecter un utilisateur avec l’authentification avec Google**

Authentifiez-vous auprès de Firebase à l'aide de l'objet fournisseur Google. Vous pouvez inviter vos utilisateurs à se connecter avec leur compte Google soit en ouvrant une fenêtre contextuelle, soit en les redirigeant vers la page de connexion.

Pour vous connecter avec une fenêtre pop-up, appelez **signInWithPopup**:

Ensuite, vous pouvez également récupérer le jeton OAuth du fournisseur Google en appelant **getRedirectResult** lors du chargement de votre page

const googleAuthProvider = new firebase.auth.GoogleAuthProvider()

// Login With Google Params

    googleAuthProvider.addScope('https://www.googleapis.com/auth/contacts.readonly');

//handle login with Facebook

    function handleGoogleLogin(e){

        e.preventDefault()

        auth.signInWithPopup(googleAuthProvider)

            .then((result) => {

                // The signed-in user info.

                const user = result.user;

                // This gives you a Facebook Access Token. You can use it to access the Facebook API.

                const credential = googleAuthProvider.credentialFromResult(result);

                const accessToken = credential.accessToken;

            })

            .catch((error) => {

                // Handle Errors here.

                console.log(error)

            });

    }

**Connecter un utilisateur avec l’authentification avec Facebook**

Vous pouvez permettre à vos utilisateurs de s'authentifier auprès de Firebase à l'aide de leurs comptes Facebook en intégrant Facebook Login dans votre application. Vous pouvez intégrer la connexion Facebook soit en utilisant le SDK Firebase pour effectuer le flux de connexion, soit en effectuant manuellement le flux de connexion Facebook et en transmettant le jeton d'accès résultant à Firebase.

Les étapes à suivre on doit créer une application dans le site de Facebook développer

Url : <https://developers.facebook.com/>

Après on colle un URI de Firebase dans notre app sur Facebook qu’on créer afin de lié notre projet Firebase avec Facebook

Une image contenant texte

Description générée automatiquement

Après on appelle notre Facebook Provider

const FacebookAuthProvider = new firebase.auth.FacebookAuthProvider()

//handle login with Facebook

    function handleFacebookLogin(e){

        e.preventDefault()

        auth.signInWithPopup(FacebookAuthProvider)

            .then((result) => {

                // The signed-in user info.

                const user = result.user;

                // This gives you a Facebook Access Token. You can use it to access the Facebook API.

                const credential = FacebookAuthProvider.credentialFromResult(result);

                const accessToken = credential.accessToken;

            })

            .catch((error) => {

                // Handle Errors here.

                console.log(error)

            });

    }

**Déconnecter l’utilisateur**

Pour déconnecter l’utilisateur on utilise la fonction signOut() après on redirige l’utilisateur dans la page login

// handle Log Out

    function handleLogOut(e){

        e.preventDefault()

        auth.signOut()

            .then(() => {

                history.push('/login')

            })

            .catch(error => {

                console.log("ERROR", error.message)

            })

    }

**Modifier le mot de passe de l’utilisateur**

Si un utilisateur veut modifier son mot de passe il doit se connecter dans l’application Yfiles et se rendre dans la page profil :

// Prompt the user to re-provide their sign-in credentials

                user.reauthenticateWithCredential(credential).then(function() {

                    user.updatePassword(state.newPassword).then(function() {

                        // Update successful.

                        console.log("everything OKEY")

                        setError( "" )

                        setSuccess( "Your password is updated successfully" )

                        setAlertStyleSuccess( "alert alert-success" )

                        setAlertStyleError( "alert alert-danger d-none" )

                    }).catch(function(error) {

                        console.log("ERROR updatePassword !", error.code)

                    });

                }).catch(function(error) {

                    console.log("ERROR reauthenticateWithCredential !", error.code)

                });

**Réinitialisation du mot de passe par email**

Si l’utilisateur a oublié son mot de passe y’a la possibilité de le récupéré via an email send de la par de google on utilisant la fonction **sendPasswordResetEmail()**

// handle update password with email send

    function handleUpdateUserPassword(e){

        e.preventDefault()

        auth.languageCode = 'fr';

        auth.sendPasswordResetEmail(email).then(function() {

        // Email sent.

            console.log("email send")

            setSuccess( "An email has been sent, Please check you email address in order to update your password" )

            setAlertStyleSuccess( "alert alert-success" )

        }).catch(function(error) {

        // An error happened.

            console.log("Error!", error)

        });

    }

**Réinitialisation du mot de passe par email**

Dans la page profil de l’utilisateur peut supprimer son compte avec cette fonction **user.delete()**

// delete user

    function handleDeleteUserAccount(e){

    e.preventDefault()

    const user = auth.currentUser

    var r = window.confirm("You sure that you want delete your account, you risk to loose all your folders and files !")

    if(r === true){

        user.delete().then(function() {

        // User deleted.

        }).catch(function(error) {

        // An error happened.

            console.log(error)

            alert("Please try to log out and login in again in order to do this operation")

        });

    }

    }

App Mobile

Nous avons utilisé le package web view pour react native avec cette commande

Nous avons créé un projet en react native après ca on a héberger notre application sur Firebase pour avoir un URL qu’on va donner comme URI au balise WEBVIEW sur l’application react native

**$ npm install react-native-webview**

Les webview c’est quoi ?

Un moteur de navigateur contenu dans une application. Le WebView permet au programmeur d'écrire la majeure partie de l'application en utilisant HTML, JavaScript et CSS, les outils de programmation Web standard.