**NOTES DU PROJET (CAMPUS INSPECT)**

1. UTILISATION DE FIREBASE
2. Nous devons commencer par importer firebase et les fonctionnalités que nous allons utiliser et ensuite les initialiser

 <script type="module">

*// Importer les fonctions nécessaires depuis les SDK de Firebase*

*import* { initializeApp } *from* "https://www.gstatic.com/firebasejs/10.14.0/firebase-app.js"; *// Pour initialiser l'application Firebase*

*import* { getAnalytics } *from* "https://www.gstatic.com/firebasejs/10.14.0/firebase-analytics.js"; *// Pour les analyses*

*import* { getFirestore } *from* "https://www.gstatic.com/firebasejs/10.14.0/firebase-firestore.js"; *// Pour utiliser Firestore*

*import* { getStorage, ref, uploadBytes, getDownloadURL } *from* "https://www.gstatic.com/firebasejs/10.14.0/firebase-storage.js"; *// Pour utiliser Storage*

*// Configuration de votre application Firebase*

const firebaseConfig = {

  apiKey: "AIzaSyDibbuBJ2p88T26P0BAB-o\_exunK0GYFdA", *// Clé API de votre projet*

  authDomain: "inspecteur-de-classes.firebaseapp.com", *// Domaine d'authentification*

  projectId: "inspecteur-de-classes", *// ID de votre projet*

  storageBucket: "inspecteur-de-classes.appspot.com", *// Bucket de stockage pour les fichiers*

  messagingSenderId: "572661846292", *// ID de l'expéditeur de messages*

  appId: "1:572661846292:web:aeb0374db2d414fef9f201", *// ID de votre application*

  measurementId: "G-NVN5GERDV6" *// ID de mesure pour les analyses*

};

*// Initialiser Firebase avec la configuration fournie*

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app); *// Initialiser les analyses*

*// Initialiser Firestore (base de données de Firebase)*

const db = getFirestore(app); *// Maintenant, Firestore est prêt à être utilisé*

*// Initialiser Firebase Storage (pour stocker des fichiers)*

1. const storage = getStorage(app); *// Utilisez getStorage pour initialiser le stockage*
2. Ajout de données dans firestore

*// Importez les fonctions nécessaires depuis Firebase*

*import* { getFirestore, collection, addDoc, getDocs } *from* "https://www.gstatic.com/firebasejs/10.14.0/firebase-firestore.js";

*// Assurez-vous que Firebase est déjà initialisé dans votre fichier HTML*

const db = getFirestore(); *// Assurez-vous que cela soit défini après l'initialisation de Firebase*

async function addUser() {

*try* {

        const docRef = *await* addDoc(collection(db, "users"), {

            first: "Ada",

            last: "Lovelace",

            born: 1815

        });

        console.log("Document écrit avec l'ID : ", docRef.id);

    } *catch* (e) {

        console.error("Erreur lors de l'ajout du document : ", e);

    }

}

1. **Récupérer les données enregistrées dans une collection (Sur firestore)**

async function getElements() {

const querySnapshot = *await* getDocs(collection(db, "users"));

querySnapshot.forEach((*doc*) => {

  console.log(`${*doc*.id} => ${*doc*.data().born}`);

});

}

1. **Sauvegarder un fichier(Image) dans Firestore storage**

*// Fonction pour télécharger un fichier*

async function uploadFile(*file*) {

*// Créer une référence au lieu où le fichier sera stocké*

  const fileRef = ref(storage, 'uploads/' + *file*.name); *// Référence au fichier avec son nom*

*try* {

*// Télécharger le fichier à l'emplacement spécifié*

    const snapshot = *await* uploadBytes(fileRef, *file*); *// Mettre le fichier dans le stockage*

    console.log('Uploaded a file!', snapshot); *// Afficher un message de succès*

*// Obtenir l'URL du fichier téléchargé*

    const downloadURL = *await* getDownloadURL(fileRef); *// Récupérer l'URL pour accéder au fichier*

    console.log('File available at', downloadURL); *// Afficher l'URL dans la console*

  } *catch* (error) {

    console.error('Upload failed:', error); *// Afficher une erreur si le téléchargement échoue*

  }

}

*// Ajouter un écouteur d'événements pour le bouton de téléchargement*

document.getElementById('uploadButton').addEventListener('click', () => {

*// Obtenir l'élément d'entrée de fichier*

  const fileInput = document.getElementById('fileInput');

  const file = fileInput.files[0]; *// Récupérer le premier fichier sélectionné*

*// Vérifier si un fichier a été sélectionné*

*if* (file) {

    uploadFile(file); *// Appeler la fonction pour télécharger le fichier*

  } *else* {

    console.error('No file selected!'); *// Afficher un message d'erreur si aucun fichier n'est sélectionné*

  }

});

1. **Sauvegarder les informations d’un étudiant sur Firestore et sa photo sur Storage**

*// Fonction pour inscrire un étudiant*

async function registerStudent(*studentInfo*, *file*) {

*try* {

*// 1. Upload de la photo dans Firebase Storage*

    const storageRef = ref(storage, 'photos/' + *file*.name); *// Référence du fichier photo*

    const snapshot = *await* uploadBytes(storageRef, *file*); *// Uploader la photo*

    const downloadURL = *await* getDownloadURL(storageRef); *// Récupérer l'URL de la photo*

*// 2. Stocker les informations de l'étudiant dans Firestore, y compris l'URL de la photo*

    const studentData = {

      name: *studentInfo*.name,      *// Nom de l'étudiant*

      email: *studentInfo*.email,    *// Email de l'étudiant*

      photoURL: downloadURL        *// URL de la photo*

    };

*// Ajouter l'étudiant dans la collection "students" de Firestore*

*await* addDoc(collection(db, "students"), studentData);

    console.log("Étudiant inscrit avec succès !");

  } *catch* (error) {

    console.error("Erreur lors de l'inscription de l'étudiant :", error);

  }

}

*// Exemple d'utilisation*

document.getElementById('registerButton').addEventListener('click', () => {

  const name = document.getElementById('nameInput').value; *// Récupérer le nom*

  const email = document.getElementById('emailInput').value; *// Récupérer l'email*

  const fileInput = document.getElementById('fileInput');

  const file = fileInput.files[0]; *// Récupérer la photo*

  const studentInfo = { name, email }; *// Créer l'objet des infos de l'étudiant*

*if* (file) {

    registerStudent(studentInfo, file); *// Appeler la fonction d'inscription*

  } *else* {

    console.error('Aucune photo sélectionnée !');

  }

});

1. **Récupérer les données de l’étudiant (Avec sa photo)**

*/ Fonction pour récupérer et afficher les informations d'un étudiant*

async function displayStudentInfo(*studentId*) {

*try* {

*// Obtenir les informations de l'étudiant à partir de Firestore*

    const studentDoc = *await* getDoc(doc(db, "students", *studentId*));

*if* (studentDoc.exists()) {

      const studentData = studentDoc.data(); *// Récupérer les données*

      console.log("Nom :", studentData.name);

      console.log("Email :", studentData.email);

      console.log("Photo URL :", studentData.photoURL);

*// Afficher la photo en utilisant l'URL*

      document.getElementById('studentPhoto').src = studentData.photoURL;

    } *else* {

      console.log("Aucun étudiant trouvé avec cet ID");

    }

  } *catch* (error) {

    console.error("Erreur lors de la récupération des infos de l'étudiant :", error);

  }

}

*// Exemple d'utilisation pour afficher les informations d'un étudiant*

displayStudentInfo("studentId123"); *// Remplacer par l'ID de l'étudiant*