Membre du groupe :

Abdourahmane Diouf

Beni Miansi Pakhe

Rapport sur le Projet de Gestion des Contacts

Introduction :

Le présent rapport décrit le projet de développement d'une application de gestion des contacts. L'objectif principal de ce projet était de créer une application permettant aux utilisateurs de saisir, afficher, modifier et supprimer des contacts à partir d'un fichier XML. Le projet a été réalisé en utilisant les technologies HTML, CSS, JavaScript et XML.

Fonctionnalités Clés :

Formulaire d'ajout de contact : Les utilisateurs peuvent saisir les détails d'un nouveau contact via un formulaire convivial.

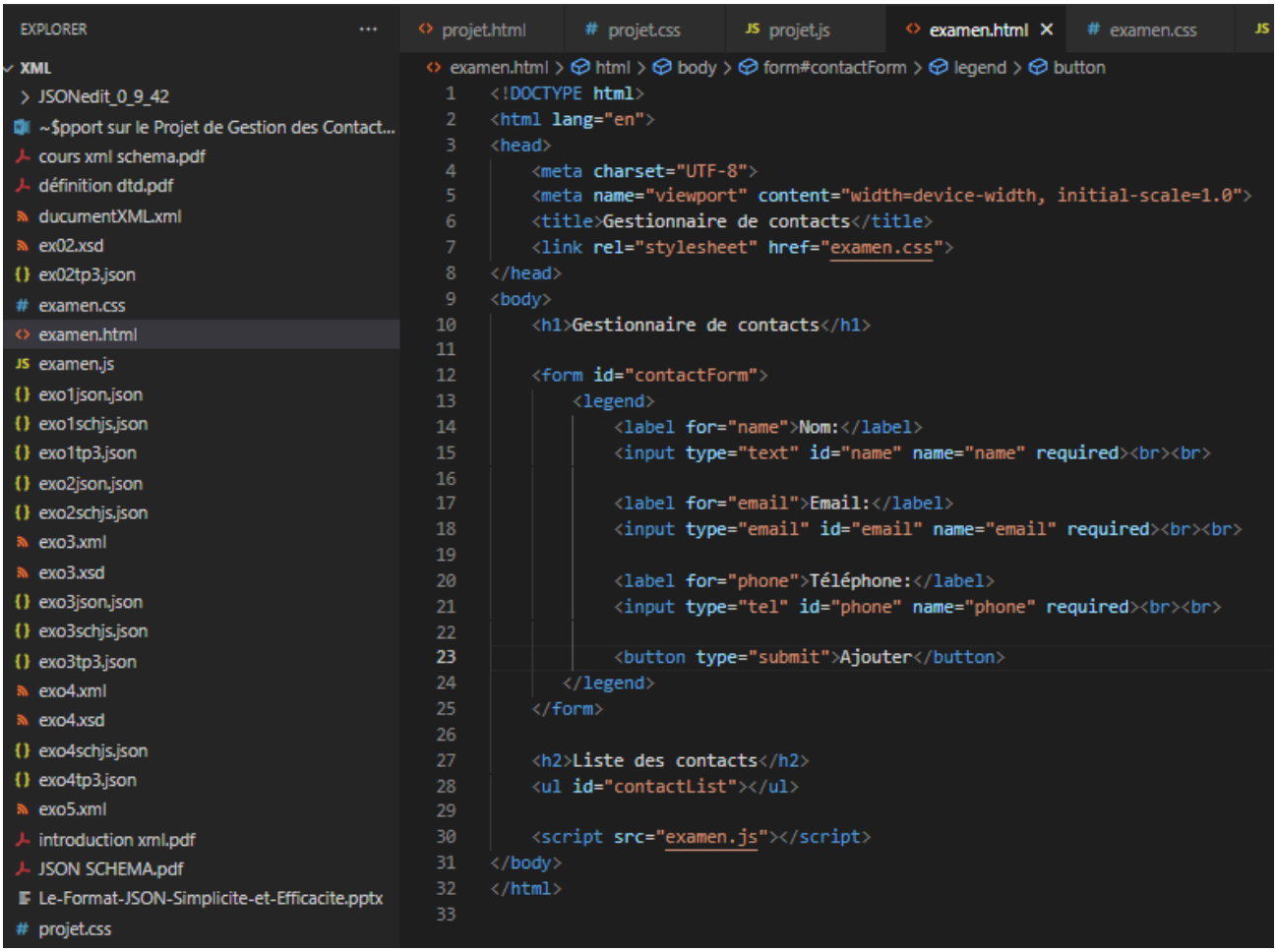
Affichage de la liste des contacts : Les contacts existants sont affichés dans une liste pour une consultation facile.

Modification de contact : Les utilisateurs peuvent modifier les détails d'un contact existant directement depuis l'interface.

Suppression de contact : Les utilisateurs peuvent supprimer un contact de la liste en un clic.

Étapes de Réalisation :

Conception du formulaire : Une interface utilisateur a été développée pour permettre aux utilisateurs de saisir les détails d'un nouveau contact de manière intuitive.



Gestionnaire de contacts

Nom : sonko

Courriel : abdourahmanelama@2gmail.com

Téléphone : 782340921

Ajouter

Lecture du fichier XML : Le fichier XML contenant les contacts a été lu et les contacts ont été affichés dans une liste sur l'interface utilisateur.

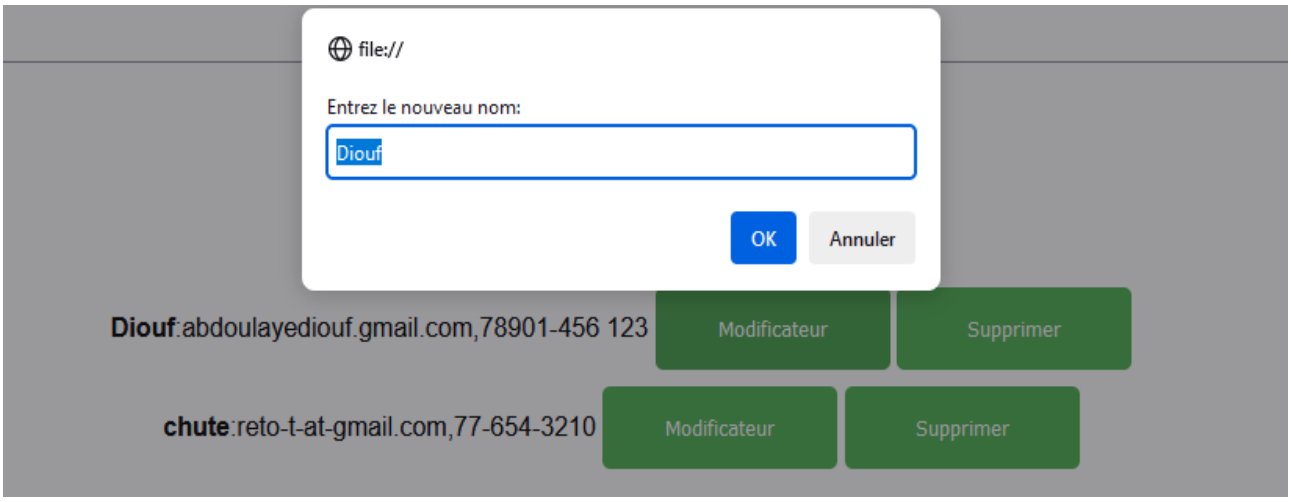
```
// Lecture du fichier XML (simulation)
// Supposons que le fichier XML a déjà été chargé et stocké dans une variable contactsXML

// Simulation des contacts à partir du fichier XML
const contactsXML = `<contacts>
                        <contact>
                            <name>Diouf</name>
                            <email>abdoulayediouf@gmail.com</email>
                            <phone>78901-456 123</phone>
                        </contact>
                        <contact>
                            <name>fall</name>
                            <email>fallala@gmail.com</email>
                            <phone>77-654-3210</phone>
                        </contact>
                    </contacts>`;

const parser = new DOMParser();
const xmlDoc = parser.parseFromString(contactsXML, 'text/xml');
const contacts = xmlDoc.getElementsByTagName('contact');
```

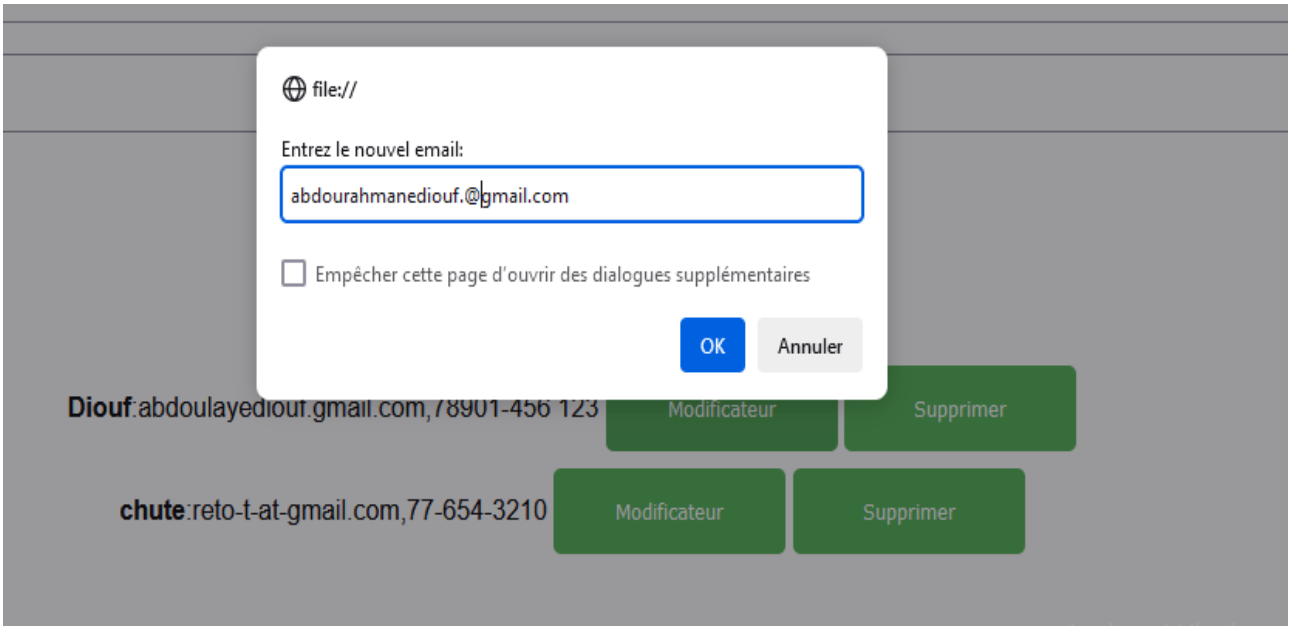
Ajout de contact : Un mécanisme a été mis en place pour permettre l'ajout de nouveaux contacts au fichier XML à partir du formulaire d'ajout.

```
// Ajout de contact
contactForm.addEventListener('submit', function(event) {
    event.preventDefault();

    const name = contactForm.elements['name'].value;
    const email = contactForm.elements['email'].value;
    const phone = contactForm.elements['phone'].value;

    addContact(name, email, phone);

    // Effacer les champs du formulaire après l'ajout
    contactForm.reset();
});
```

Modification de contact : Les utilisateurs ont été autorisés à modifier les détails d'un contact existant, avec mise à jour simultanée du fichier XML.

```
    // Modification de contact
contactList.addEventListener('click', function(event) {
    if (event.target.classList.contains('edit')) {
        const li = event.target.parentElement;
        const nameElement = li.querySelector('.name');
        const emailElement = li.querySelector('.email');
        const phoneElement = li.querySelector('.phone');

        const newName = prompt('Entrez le nouveau nom:', nameElement.textContent);
        const newEmail = prompt('Entrez le nouvel email:', emailElement.textContent);
        const newPhone = prompt('Entrez le nouveau téléphone:', phoneElement.textContent);

        if (newName && newEmail && newPhone) {
            nameElement.textContent = newName;
            emailElement.textContent = newEmail;
            phoneElement.textContent = newPhone;

            // Mettre à jour le fichier XML (simulation)
            const contact = li.getElementsByTagName('contact')[0];
            contact.getElementsByTagName('name')[0].textContent = newName;
            contact.getElementsByTagName('email')[0].textContent = newEmail;
            contact.getElementsByTagName('phone')[0].textContent = newPhone;
        }
    }
});

// Ajout de boutons d'édition aux contacts existants
for (let i = 0; i < contacts.length; i++) {
    const name = contacts[i].getElementsByTagName('name')[0].textContent;
    const email = contacts[i].getElementsByTagName('email')[0].textContent;
    const phone = contacts[i].getElementsByTagName('phone')[0].textContent;
    addContact(name, email, phone);
}

function addContact(name, email, phone) {
    const li = document.createElement('li');
    li.innerHTML = `<strong class="name">${name}</strong>: <span class="email">${email}</span>, <span class="phone">${phone}</span>
    <button class="edit">Modifier</button> <button class="delete">Supprimer</button>`;
    contactList.appendChild(li);
}
```





Suppression de contact : Une fonctionnalité a été implémentée pour permettre aux utilisateurs de supprimer des contacts de la liste, avec suppression correspondante dans le fichier XML.

```
    // Suppression de contact
    contactList.addEventListener('click', function(event) {
        if (event.target.classList.contains('delete')) {
            event.target.parentElement.remove();
        }
    });
});
```

Technologies Utilisées :

HTML/CSS pour l'interface utilisateur.

JavaScript pour la logique côté client (lecture, ajout, modification, suppression de contacts).

XML pour stocker les données des contacts.

Conclusions :

Le projet de gestion des contacts a été mené à bien avec succès. L'application développée répond aux exigences spécifiées et offre une expérience utilisateur fluide et intuitive pour la gestion des contacts. Les technologies utilisées ont été choisies de manière appropriée pour répondre aux besoins du projet. Des tests ont été effectués pour assurer le bon fonctionnement de l'application et des correctifs ont été apportés en cas de besoin.

En conclusion, ce projet a permis de mettre en pratique divers concepts de développement web et de manipulation de données XML. Il a également fourni une expérience précieuse dans la planification, la conception et la mise en œuvre d'une application fonctionnelle dans un environnement client-serveur.